

Articles

Beyond the Magic to the Mechanism: Computers, Materiality, and What It Means for Records to Be “Born Digital”



CIARAN B. TRACE

RÉSUMÉ Au courant des trente-cinq dernières années, les archivistes ont travaillé à l'établissement d'un ensemble de savoirs et d'expertise afin de gérer et de conserver les documents créés dès l'origine en format numérique. Toutefois, ce qui n'apparaît pas dans la recension des écrits, c'est un texte qui explique comment les parties constituan-tes d'un ordinateur travaillent ensemble pour créer, gérer et stocker ces objets numé-riques qui pourraient éventuellement avoir une valeur de longue durée ou une valeur continue. Comprendre et travailler avec le matériel créé numériquement nous oblige à ouvrir la boîte noire de l'ordinateur et à nous familiariser avec l'équipement informa-tique, les logiciels d'application, les logiciels d'exploitation et les micrologiciels, tout comme nous avons jadis été obligés de nous familiariser avec les technologies liées à l'écriture, à l'archivage et à la conservation. Avant tout, ce texte joue un rôle sembla-ble à celui de la micro-ethnologie, alors que les actions de créer et de sauvegarder un document numérique sont enracinées dans un examen de la séquence des technologies utilisées pour que ces actions soient complétées. Ce faisant, ce texte expose les traces matérielles qui résultent de l'interaction entre les êtres humains et les ordinateurs dans la création d'un document.

ABSTRACT Over the past thirty-five years, archivists have worked to establish bodies of knowledge and expertise to help manage and preserve born-digital records. What has been missing from the archival literature is a foundational piece that explicitly explores how the components of a computer work together to create, manage, and store those digital objects, which are later deemed to have long-term or enduring value. Un-derstanding and working with born-digital material requires us to open the computer's black box and become as familiar with computer hardware, application software, sys-tem software, and firmware as we have become with earlier writing, recording, and storage technologies. At its heart, this article takes on a role similar to that of a micro-ethnography, whereby the actions to create and save a digital document are grounded in an examination of the sequence of technologies utilized to bring such actions to fruition. In the process, this article brings to life the material traces that are left behind as human and computer interact to create the record.

Introduction

According to social scientist Sherry Turkle, our way of understanding and interacting with computers has changed over the past forty years.¹ In the early days of the personal computer in the 1970s and early 1980s, peoples' relationship with computers, as characterized later by Turkle, adhered to a modernist view of reality – in which reality was seen as “‘linear’, ‘logical’, ‘hierarchical’, and by having depths that could be plumbed and understood.”² In such a modernist perspective, computers were designed to be “open” and “transparent,” “potentially reducible to their underlying components.”³ People were encouraged to see and understand the inner workings of the computer, to look “beyond the magic to the mechanism.”⁴ This modernist view of reality is perhaps best brought to life in the persona of the computer hobbyist with his/her goal of “reductive understanding” – “open the box, trust what you can see, and analyze completely.”⁵ Turkle sees the introduction of the Macintosh computer in the 1980s as heralding a new relationship between people and computers; “a way of thinking that put a premium on surface manipulation and working in ignorance of the underlying mechanism.”⁶ Rather than encouraging the user to “open the hood, and see inside,”⁷ the design of the Mac presented the user with “a scintillating surface on which to float, skim, and play.”⁸ The postmodern view is that of the computer as “opaque technology,” whereby people take the computer at “interface value,” and whereby the computer screen becomes a world unto itself.⁹

The struggle between the modern and the postmodern – the technological aesthetic versus the aesthetics of simulation, as Turkle calls it – is being played out in the archival profession, even if we do not necessarily articulate it in this manner. In coming to grips with fundamental changes to the archival profession that are being brought about by technology, archivists are trying to figure out what the appropriate nature and level of engagement should be with computer systems. Should archivists stay at the surface level – using computers as a tool to help make collections more available and accessible but eschewing any deeper knowledge of how computers and computer systems work? Or should archivists dive below the surface in order to understand the inner workings of the computer – knowledge that could help archivists understand, at the most granular level, the fundamental nature of born-digital material? The latter be-

1 Sherry Turkle, *Life on the Screen: Identity in the Age of the Internet* (New York, 1995).

2 *Ibid.*, p. 17.

3 *Ibid.*, p. 23.

4 *Ibid.*

5 *Ibid.*, p. 43.

6 *Ibid.*, p. 35.

7 *Ibid.*

8 *Ibid.*, p. 34.

9 *Ibid.*, p. 23.

ing the case, archivists would certainly need what Richard Pearce-Moses calls “under-the-hood IT skills.”¹⁰

Recent efforts indicate that the archival profession is increasingly embracing both aesthetics. Nowhere is this more apparent than in the recently revised Society of American Archivists’ *Guidelines for a Graduate Program in Archival Studies* (GPAS).¹¹ In GPAS, the technological aesthetic is evident in the requirement that students, as part of the core archival knowledge area, understand digital records and access systems (which includes knowledge of “file formats, media types, and complex information technologies for the creation, maintenance, use, and preservation of all types of records”). According to GPAS, the broader field of information technology complements core areas of archival knowledge, and in this “complementary knowledge” arena we see in play both the technological aesthetic and the aesthetics of simulation, with students expected to understand human/computer interaction (HCI) in order to design and develop effective systems for users, as well as having skills in areas such as database design and management, the creation and design of websites, and computer programming.¹²

The revisions to GPAS highlight that the archival workplace and archival work processes are changing. Archivists are drawing from the concepts, tools, and techniques of both computer science and digital forensics in order to

10 Richard Pearce-Moses, “A Bridge to the Future: Committing Intentional Acts of Memory” (paper presented at the Society of Southwestern Archivists, Louisiana, 2005), <http://www.lib.az.us/diggovt/documents/pdf/Bridge.pdf> (accessed on 13 August 2011).

11 Society of American Archivists, *Guidelines for a Graduate Program in Archival Studies* (Chicago, 2011), <http://www2.archivists.org/gpas> (accessed on 13 August 2011).

12 It must be acknowledged that a tension exists between educators and practitioners about the perceived proper role of graduate education – whether the curriculum of a graduate archival education program should be built around knowledge, scholarship, and disciplinary criticism, or around skills and competencies. GPAS explicitly distinguishes between education (the focus of graduate archival education programs) and training (characterized as “building skills or acquiring practical knowledge according to a replicable pattern, or on developing a specialization in certain areas”). Richard Cox has stated that, “skills and competencies are not what we should be teaching in a university classroom,” calling skills and competencies “the province of workshops, institutes, or in-service training programs.” He has also warned against technology “training,” which suggests to him “a limiting to skills inappropriate for graduate education.” As an archival educator, I would argue that the goal is not to ignore the skills and competencies valuable to the digital realm (after all, as Cox says, “introducing students to archival knowledge requires introducing them to technology, if only because technology and recordkeeping are intertwined”), but rather to situate those skills and competencies within a larger discourse. Taking a broader approach encourages students to think critically about the issues involved in managing records and therefore to be more flexible in the face of rapid changes in technology. The role of graduate archival education is also to link back technology and technological issues to fundamental concerns and scholarship about records, recordkeeping, and record-making. See Richard J. Cox, “Are There Really New Directions and Innovations in Archival Education?” *Archival Science* 6 (2006), pp. 247–61.

understand, access, and preserve born-digital records that come into the archive on contemporary and legacy computers and media.¹³ Computers and computer systems are integral to the accessioning, arrangement, description, storage, and preservation of digital collections. Archivists are also turning to technology in order to design tools to help users access and use archival collections in a digital environment. Given these facts, archivists today need to meet the challenge of their workplace by adopting a dual approach, one that is both modern and postmodern. Both ways have merit; both are necessary. This article, however, does not attempt to address the entire scope of this broad issue. Instead, it seeks to help redress a “modern” oversight: the general lack of archival literature that specifically supports archivists in the role of computer hobbyist.

By diving into the inner workings of the computer and the computer system, archivists will achieve two things. First, archivists can more fully embrace sophisticated models of how digital records exist within computer systems. Kenneth Thibodeau’s tripartite model of digital objects, which takes archivists beyond our traditional understanding of records as conceptual objects (records as they are presented and viewed by the user) to an “under-the-hood” view that shows that digital records also exist as physical objects (things that are inscribed on a medium) and as logical objects (objects that can be recognized and processed by programs and software), is a case in point.¹⁴ As computer hobbyists we can unpack Thibodeau’s notion of the record as physical, logical, and conceptual object, and in the process more fully comprehend how the properties of digital records are forged in the relationship between computer hardware, operating system, and application software.¹⁵ Second, archivists will be better equipped to understand the tools and processes of digital forensics that can help in the acquisition and preservation of born-digital records. Recent work by Matthew Kirschenbaum, Richard Ovenden, and Gabriela Redwine on the intersection of digital forensics and archives demonstrates that archivists must be able to access and evaluate the different layers of abstraction that constitute the computer system (“from the pixels on the screen to the magnetic

13 For a detailed look at the application of digital forensics tools and technologies to the archival environment, see Matthew G. Kirschenbaum, Richard Ovenden, and Gabriela Redwine, *Digital Forensics and Born-Digital Content in Cultural Heritage Collections*, CLIR Publication 149 (Washington, DC: Council on Library and Information Resources, 2010). For a look at how diplomatics is being merged with digital forensics, see Luciana Duranti, “From Digital Diplomats to Digital Records Forensics,” *Archivaria* 68 (Fall 2009), pp. 39–66.

14 Kenneth Thibodeau, “Overview of Technological Approaches to Digital Preservation and Challenges in Coming Years,” *The State of Digital Preservation: An International Perspective*, CLIR Publication 107 (Washington, DC: Council on Library and Information Resources, 2002).

15 Ibid.

traces on the disk”) in order to preserve the records stored within it.¹⁶ Such work involves understanding what is contained on a computer’s hard drive so that the archivist knows what is captured through a disk image (an exact copy) of the device; understanding how file systems work so that the archivists can capture files from various storage media; and understanding the function of a computer’s registry so that archivists do not unknowingly make changes to important recordkeeping metadata.

Explanations of the operations of computer hardware and software abound. Indeed, there are many books, online articles, and videos that explain this subject in as much detail as anyone could want to know. What is new in this article is that a statement is being made that this “modern” knowledge should be made more available to archivists in the form of our own writing.¹⁷ Indeed, if archival texts include information about the technical processes by which a primary support layer, a binder layer, and an imaging layer work together to form a photograph, or if we need to know how the sound of a performer, a mouthpiece with a diaphragm, a cutter, and a wax cylinder work together to create a sound recording, so too should archival writing present an in-depth examination of the processes by which hardware and software work together to form a digital record.

At its heart, this article acts almost like a form of microethnography,¹⁸ whereby the actions to create and save a digital document on a personal desktop computer (PC) are grounded in a detailed examination of the key components of the computer system that bring such actions to fruition.¹⁹ This journey beyond the magic to the mechanism harks back to Turkle’s modernist view of understanding the mechanisms of life, where computers are studied as material artifacts and what Kirschenbaum calls “objects in the world.”²⁰ In looking at the birth of a digital record, this journey starts with an examination of the concept of the computer, providing an analogy to help archivists understand its basic functions. The main body of this article opens with an examination of the language in which computers process and store data and instructions, and then moves on to examine computer hardware and software functions. This exploration into the various levels of abstraction of a computer system is explicitly

16 Kirschenbaum, Ovensen, and Redwine, *Digital Forensics*, p. 7.

17 In the 1980s archivists had started to incorporate information about how computers worked from a number of the Society of American Archivists’ Basic Manual Series. See for example H. Thomas Hickerson, *Archives and Manuscripts: An Introduction to Automated Access* (Chicago, 1981). Another example is Margaret L. Hedstrom, *Archives and Manuscripts: Machine-Readable Records* (Chicago, 1984).

18 Microethnography is a qualitative research method typically used to closely analyze discrete events and small moments of human activity.

19 I am limiting my discussion to personal computers that run the Windows Operating System.

20 See Turkle. Matthew Kirschenbaum, *Mechanisms: New Media and the Forensic Imagination* (Cambridge, 2008), p. 2.

linked to an examination of how these component parts affect the documentary product. In this manner, the article sheds light on the technological context that brings digital records to life.

In the Beginning – On Bits, Binary, and Boole

In order to understand at a very general level how computers work, an analogy may be useful here.²¹ Today's personal computer is like a self-contained office park composed of numerous buildings belonging to one company, including a main building and a company headquarters, and many different employees, each with specific functions. The company's main building is home to numerous clerks, whose job it is to carry out calculations and apply their reasoning skills to company data at the request of those higher up in the organization. The company headquarters building is home to middle management. It also houses the company's records centre, composed of filing cabinets that store large volumes of company data needed for extended periods of time. Running the records centre and managing the work of the clerks are some of the tasks of middle management. When the boss needs information, a clerk is sent a specific message that is temporarily stored in a special section of the clerk's own pigeonhole filing system. The message tells the clerk where the cards containing the necessary data are located in the clerk's pigeonhole filing system, as well as what calculations to perform on the data. The clerk responds by retrieving the appropriate cards, performing the calculations on the data, and then putting the result back on another card in the pigeonhole filing system. If the boss's query involves information that is not in the clerk's immediate filing system, the clerk will check his/her desk where copies of the most frequently used cards are often left. On occasion the boss may need to instruct a secretary to retrieve cards from the records centre in the company headquarters or from a temporary records storage room in the main building. The instructions and the cards are then placed in the clerk's own filing system, where the query can be processed and the result put back in the appropriate location.

In this analogy, the company's main building represents the computer's motherboard. The clerk represents that part of the motherboard called the central processing unit (CPU). The clerk's personal filing system is analogous to high-speed storage areas within the CPU called registers; the clerk's desk is analogous to cache memory; and the temporary records storage room in the main building equates to the computer's main memory (random access memory

21 This analogy is based on the file clerk model of computing created by physicist Richard Feynman and subsequently refined by computer engineer Jon Stokes. Richard Feynman, *Feynman Lectures on Computation*, eds. Anthony J. G. Hey and Robin W. Allen (Reading, MA, 1996); Jon Stokes, *Inside the Machine: An Illustrated Introduction to Microprocessors and Computer Architecture* (San Francisco, 2007).

[RAM]). The company headquarters is analogous to the hard drive of the computer, which stores data, information about the computer system, the operating system, and any application software installed on the computer. The boss’s message represents the logic, arithmetic, and memory access instructions that form part of a computer program. Middle management is the equivalent of the computer’s operating system.

This analogy speaks to the fact that the computation, calculation, storage, and manipulation of information that is done by computers today is work that was mainly carried out by humans prior to the Second World War. To all intents and purposes, the computer has subsumed, automated, and augmented certain work processes (particularly ones that are repetitive and rules-based) that were once the domain of file clerks and people employed to do calculations – so-called human computers.²² Not only has this technology annexed human work and work processes, the computer has also integrated and re-imagined the attendant recordkeeping paraphernalia of records, files, file folders, file cabinets, and file rooms – all this contained within an object that has an increasingly small footprint.

What this analogy also shows is that, at its very heart, a computer is simply a machine that “reads, modifies, and writes sequences of numbers” in order to achieve a desired result.²³ In effect, today’s personal computer is a programmable, electronic device that performs four main functions: 1) it receives data or input from devices (such as a keyboard and mouse); 2) it processes data by automatically performing a sequence of arithmetic or logical tasks; 3) it retains data through the use of various memory and storage technologies; and 4) it outputs data (via devices such as monitors and printers).²⁴ In performing these functions, computers are designed around the notion of levels of abstraction. According to this concept, a particular part of a computer system will take orders from a level of the computer system that is higher than itself and, in turn, any given part of a computer system will give orders to a level below it. The notion of abstraction also means that the particular details of the lower layers are largely hidden from the higher layers of a computer system. From the lowest to the highest, these levels, each with its own language (referenced in brackets), include: 1) a logic gate level (Boolean algebra); 2) a microcode level (microinstructions); 3) an instruction set architecture level (machine language); 4) an operating system level (operating system calls); 5) an assembly level (assembly

22 Martin Campbell-Kelly and William Aspray, *Computer: A History of the Information Machine* (New York, 1996).

23 Stokes, p. 4.

24 In the strictest sense, memory consists of electronic components that hold data and instructions for computing on a temporary basis. The term storage, on the other hand, is used to refer to devices and media that are non-volatile and that are not directly accessible by the central processing unit.

language); 6) a higher-order language level (machine-independent programming language); and 7) an application software level (language is dependent on the application).²⁵

When a person sits down in front of a personal computer to create and save a digital record, the interaction between human and computer is taking place at the highest level of abstraction – the level of the application software. However, to understand how the record as it appears on the computer screen comes into being, we have to delve deeper into the computer system by looking at the record as both a physical and a logical object. As a physical object, a digital record is “simply an inscription of signs on a medium.”²⁶ However, to fully understand the record as a physical object, we must understand how data (and thus records) can be represented and manipulated in a system that is both electronic and digital.

Today’s computer is an electronic device made up of millions of switches that exist in one of two states (on or off). These states generally exist as electronic signals of different voltage levels. With these switches, electrical current can be routed along particular pathways, and in this manner, data and instructions (in the form of these electrical signals) can be moved, stored, and manipulated within the system. In a computer, data and instructions (programs) must be represented in a language that is capable of conveying these two states. A computer language also needs to take into account that today’s PCs are generally digital and not analogue devices and, as such, electrical signals can only be stored in memory in a fixed number of states. Personal computers therefore do not use the more familiar base-10 decimal system to represent data and instructions, but a base-2 numbering system called binary, a language that allows two mutually exclusive states to exist – on or off, represented by 1 and 0 (each 0 and 1 is a binary digit or bit). In a computer system, binary language is made material (those 1s and 0s are read, moved, and stored) in a variety of ways, including through the instantiation of an open or closed circuit, the presence or absence of an electronic charge in a transistor, and through magnetic states that can be either positive or negative. While a bit is the smallest unit of information that can be stored or manipulated in a PC, a byte (a combination of 8 bits) is the basic storage unit in the computer’s memory. Each byte contains a combination of 0s and 1s that the computer can interpret as a character, a memory address, or as part of an instruction. Using a more familiar analogy, we can say that bytes are like letters of the alphabet, while bits are like the writing strokes that make up each letter.

For personal computers, the processing of data and instructions takes place through the use of another language. Comparable to the binary system, Boolean algebra uses the notion of two fundamental expressions as a means of process-

25 J. Stanley Warford, *Computer Systems* (Sudbury, MA, 2009).

26 Thibodeau, p. 3.

ing data and instructions. Boolean language is capable of describing and manipulating logical statements and determining whether or not they are true (1) or false (0). A computer can process more complex problems by combining logical statements using logical operators such as AND, OR, and NOT; computers can make decisions using these mathematical constructs. Within a PC, Boolean logic is made possible through the use of transistors – tiny electronic devices that can amplify electronic signals and can also act as a microscopic electronic switch. When brought together, transistors form logic gates that mimic Boolean logic by performing logical operations on binary data drawn from memory cells in order to produce binary outputs. By combining logic gates, computers can be made to do such things as binary addition and multiplication, as well as create memory by storing binary digits. In effect, logic gates allow computers to take an input, process an input, and then return an output.

The Black Box – Hardware

Understanding the language in which computer data and instructions are represented and manipulated is just one part of understanding the record as a physical object. Another key element is an understanding of the physical components that are necessary for a computer to carry out the tasks (gaming, programming, word processing, etc.) that it has been given (tasks that are specified in software through a set of instructions). An understanding of computer hardware requires us to open the black box and take a look at the computer’s inner workings. While the level of abstraction at which computer hardware resides, including the motherboard and all its component parts, may seem very far removed from a discussion of digital records, it is important to keep in mind that every activity a computer carries out (including word processing and email) is deconstructed into logical and mathematical constructs, which are implemented when binary digits are run through different types of logic gates.

The black box is an apt metaphor for the computer if we continue to accept the idea that, for many people, the internal workings of a computer are unknown and seemingly unknowable.²⁷ However, the actual black box itself is, in effect, no more than a horizontal or vertical steel, aluminum, and plastic enclosure that houses the main components of a computer and protects these

27 In fact, on a PC running a Microsoft operating system, information about the computer can be found quite easily by clicking on Start → right clicking on Computer → clicking on Properties. Information in Properties includes the version of the operating system; the manufacturer and model of the computer; the manufacturer, model, and speed of the central processing unit (CPU); and the bit size of the CPU and operating system. More advanced details about a computer’s hardware can be found by looking under Control Panel → Administrative Tools → Computer Management → Device Manager. All attached hardware is listed, along with extensive details about the properties of each device.

components from dust and damage.²⁸ Its size and shape are determined by the layout (or so-called “form factor”) of the largest internal component housed in the case – the motherboard.²⁹ The power supply unit in a PC generally resides in a square metal box inside the case. One of the key jobs of this electrical system is to convert incoming, alternating current (AC) power from an electrical outlet into the lower voltage direct current (DC) power that is required in order to operate the sensitive electronic components inside the computer.

Inside the case, most of the components in a personal computer are mounted on printed circuit boards – ultra-thin boards of layered fibreglass that bring the components of an electronic circuit together. The combination of components (including microchips) and chemically etched copper pathways (used to connect the components) allows the computer to perform simple and complex operations, including the processing and transmission of information. Resistors, capacitors, and transistors are key electronic components in an electronic circuit. Resistors serve to regulate voltage on the circuit board by reducing, opposing, or restricting the flow of electrical current. Capacitors store and release electrical energy (electrons), providing a steady source of current to various components on the circuit board. As noted earlier, transistors work to amplify or boost an electrical current or voltage, and act as switches, turning voltage on or off.³⁰

The largest printed circuit board in a PC is the multi-layered motherboard. The motherboard acts as the hub or central nervous system for the computer – providing a home for the electronics that support many of the computer’s basic functions; implementing the pathways or buses that allow the computer to transfer data; supporting peripheral devices that can be built right into the motherboard (including video and sound cards); and serving as a connection point for internal (processor, memory, hard drive, etc.) as well as external components (keyboard, mouse, etc.). Essential components of the motherboard include the central processing unit, buses, chipset, basic input output system

- 28 Some computer cases include a switch that is tripped when the case is opened. The computer’s basic input output system (BIOS) (software programmed on a chip on the computer’s motherboard) can be configured so that this event is logged by the system.
- 29 If the design and colour of the computer case is such that the case seems to blend into the background so as to provide a cover for the inner workings of something that we need to know nothing about, then such use of design to dematerialize this object has been turned on its head by at least one community – case modders. Similar to the practice that emerged in the United States from the 1930s onward of hot-rodders modifying and customizing automobiles, case modification is the art and science of modifying a computer case by painting, tattooing, backlighting, or affixing logos or images to the case, or by cutting holes in the side panels to reveal the internal workings of the computer.
- 30 Capacitors are generally round and cylindrical and look like little batteries. Resistors look like tiny, long, thin soda cans with rings of colour. Transistors usually have three wires (that look like legs), which are attached to a conducting material. Not all these components can be seen on a printed circuit board, as some may be embedded in the electronics.

(BIOS), and main memory (RAM housed within memory slots).³¹

The central processing unit (CPU) is an electronic circuit that acts as the brains of the computer – having the circuitry needed to interpret and execute the calculations and instructions (whether it be data transfer, memory access, arithmetic or logical operations, or program control) that are received from the computer’s hardware components, operating system, and applications. In personal computers, the CPU is housed on an integrated circuit (a silicon chip) called a microprocessor.³² The CPU operates at a speed determined by an internal clock that sends out pulses or peaks. With each peak the CPU performs an action corresponding to an instruction or part of an instruction.³³ Typically, a CPU follows a set instruction or machine cycle: fetch (an instruction is fetched from main memory and loaded into the CPU); decode (the instruction is broken up into commands and sent to specific parts of the CPU); execute (the instruction is executed); and writeback (the result of the execution is stored in a CPU register or in main memory).³⁴ At its heart, the CPU is made up of millions of transistors that act as electronic switches. When brought together, these transistors form logic gates that allow the processor to perform a logical operation (usually a Boolean operation) on one or more binary signals (inputs) to produce an output. In this way the CPU is able to manipulate and process data at the bit level.

The main parts of the CPU are the control unit, the arithmetic and logic unit (ALU), and registers. The control unit contains circuitry that manages and directs the operations within the CPU, fetches and decodes stored program instructions, and controls the flow of instructions and data from an input device or secondary storage into and out of main memory. The ALU contains the electronic circuitry to execute arithmetic (addition, subtraction, multiplication, and division) and logical operations (including equal-to, greater-than, or less-than comparisons of letters, numbers, or special characters). Registers (a form of inbuilt memory cell made up of transistors inside the CPU) act as high-speed memory for the CPU by temporarily accepting, storing, and transferring

31 Macintosh computers use something akin to BIOS: Unified Extensible Firmware Interface (UEFI).

32 The CPU connects to the motherboard via a CPU socket or interface, and has an attached heat sink and fan to help heat dissipate.

33 The computer’s clock is a microchip on the motherboard that regulates the speed and time of the computer’s functions. The clock is generally controlled by a quartz crystal that vibrates at a particular frequency when electricity is applied to it. Current clock speeds are measured in gigahertz. The faster the clock speed, the faster the processor can execute instructions. A CPU of three gigahertz, for example, can execute approximately three billion cycles or operations per second.

34 In some computers, the CPU waits until an instruction completes all the stages of a machine cycle before moving on to the next instruction. Computers nowadays generally support faster processing through pipelining, whereby the processor starts another instruction before the first machine cycle is finished.

data and instructions while they are being processed. The number and size of each register (the so-called “word size”) affect the power and speed of the CPU. Registers are usually 32- or 64-bits wide. This means that the amount of data the CPU is able to process can contain up to 32 or 64 bits per instruction. Another form of memory, cache memory, may also be built into the CPU or located elsewhere on the motherboard. Cache memory is local, high-speed memory designed to accelerate processing by storing parts of a program or data that needs to be accessed repeatedly.

The motherboard’s bus system consists of electrical pathways or highways made up of tiny wires that move data packets and signals in the form of bits between the input and memory, memory and the CPU, the CPU and memory, and memory and output. Buses carry four types of signals throughout the computer: 1) data (the information carried between main memory, the CPU, and peripherals); 2) power (provides DC power to electronic components on the motherboard); 3) control (the signals to manage the transmission and movement of information between devices connected to the motherboard); and 4) address (carries addresses of data and instructions so that the computer knows the location of the next instruction to be executed, where data to be worked on is located, and where to put the results of an executed instruction). Data and instructions (including data transfer, data manipulation, and program control instructions) travel on buses as a series of electrical pulses or signals, each pulse representing a 1 bit or a 0 bit. The bus width (16 or 32 bit, for example) determines the number of bits that a computer can transport at a time and the number of memory locations to or from which a bus can transfer information. Each bus also has a clock speed (measured in megahertz), which equates to the number of data packets that can be sent or received per second; the higher the clock speed, the faster data can be transmitted.

The two main types of buses are the system bus (also known as the local or frontside bus), and the I/O (input/output) or expansion bus. A variety of system buses connect the processor to main memory as well as to other components that reside on the motherboard. The I/O bus connects the processor to external peripheral devices such as the keyboard, mouse, and printer. Peripheral Component Interconnect (PCI) is a common data bus standard for connecting I/O devices such as graphics or modem cards to the motherboard. The accelerated graphics port (AGP) bus provides a dedicated channel for attaching a video card to memory.³⁵

While the CPU is responsible for data processing, a number of other integrated circuits (microchips) on the motherboard (called the chipset) manage the transfer of data and instructions between the CPU and other computer com-

35 AGP also refers to the standard type of connection for video cards, and the actual expansion slot on the motherboard that accepts AGP video cards.

ponents, such as input, display, and storage devices. The chipset supports the CPU by acting as a controller – controlling the buses around the CPU so that the computer’s components can talk to each other. The chipset functions as the equivalent of a traffic cop, controlling the buses by determining which input (data, instructions, etc.) to send where and when. Generally the chipset consists of a pair of microchips called the north bridge and the south bridge. The north bridge chip is usually located in close proximity to the CPU, linking the CPU to high-speed devices such as RAM (main memory). Because the south bridge is further removed from the CPU, it is more suited to handle data transfers with lower bandwidth (generally handling input/output with peripheral devices).

Another important component on the motherboard is a software program called the BIOS (basic input/output system), which partly resides in a non-volatile ROM chip (Read-Only Memory) and partly in an EEPROM (electrically erasable programmable read-only memory) chip.³⁶ BIOS consists of built-in programs or services that are needed to boot up the computer,³⁷ control major system hardware, and act as an interface between the hardware, the processor, and the operating system. POST (Power On Self Test) consists of a series of diagnostic tests that BIOS runs when the computer is first turned on in order to check that certain essential hardware (including the CPU, memory, disk drives, graphics cards, and keyboard) is present and operating.³⁸ The BIOS then loads the interrupt handlers (a program used by the CPU to process a hardware interrupt/request for service) and device drivers (a program that controls the interaction between output devices and the operating system) from the operating system into memory. Perhaps the most important function of BIOS is that it locates the operating system when the computer is turned on and transfers control of the system to the operating system after the computer has booted up.³⁹ Setup is a built-in utility in BIOS that recognizes new hardware and allows users to configure motherboard and chipset settings, as well as the configuration of other basic system settings such as the types of drives that the BIOS will recognize.⁴⁰

36 A ROM chip, as a non-volatile form of memory, stores data even when a computer is turned off. As a form of read-only memory, ROM chips are useful for storing programs that do not change very often. Programs stored on ROM chips are known as firmware.

37 “Booting up” a computer refers to the process of loading an operating system into a computer’s main memory.

38 If BIOS detects a problem in the boot process, an error message or a series of beeps will often result. “Beep codes” have specific meanings (depending on what BIOS is installed) and can be found online.

39 When the computer is turned on, BIOS searches the first sector of the computer’s first hard disk to find a program called the Master Boot Record (MBR). The MBR reads a Master Partition Table (MPT) to see where the active partition is stored, and from there reads the first section of the active partition to find the boot loader – the program that begins loading the operating system into memory.

40 You can interact with the BIOS settings by turning on a PC and interrupting the POST. Once in the BIOS settings, you can do things such as change the boot order of the computer

Apart from the importance of understanding the role that BIOS plays in helping the computer operate, archivists may need to engage with the BIOS in a more hands-on manner. In dealing with legacy media, archivists may need to change BIOS settings in order to get a computer to recognize a legacy drive. Digital forensics work would also seem to indicate that BIOS, as a storage medium, may contain data (including system configuration information) that could be useful to archivists in documenting the larger technological context of which the records are a part.

While BIOS acts as a form of permanent memory (where any data stored will not be lost when the power to the computer is interrupted or turned off), a form of random access memory (RAM) functions as a temporary, fast-working memory for the computer. RAM stores the operating system, applications, and data that are currently in use by the computer so that the processor can access them quickly. As an electronic mechanism to store and retrieve data, RAM exists as microchips whereby the computer keeps operating system instructions, application instructions, and data before, during, and after being processed by the CPU's arithmetic and logic unit. The more RAM a computer contains, the faster the computer can process such data and applications. Generally, when a program (such as a word processing program) is launched, it is loaded (copied) from secondary storage (such as the hard drive) into RAM, and the CPU then executes one instruction at a time from RAM. RAM is often sold in stick form – with chips residing on memory modules/small circuit boards.⁴¹ These memory sticks are inserted into a slot on the motherboard.⁴²

Aside from the motherboard components and the power supply, the final key piece of hardware that will be discussed here is the internal hard drive, which stores file system administrative data, applications, the computer's operating system, and user-generated files. The hard drive is of particular interest to archivists as it is here that records and metadata about records are physically inscribed (albeit as magnetic traces). To use a more familiar analogy, a hard drive can be thought of as a filing cabinet; a computer's file system is analogous to the rules drawn up for filing that are instantiated through the physical presence of drawers in the filing cabinet; the computer's file directory is analogous to the

(if, for example, you want to boot the computer from a CD), set a power-on password for added system security, or check the BIOS settings for detailed system information (such as the brand and model number of the hard drive).

- 41 In a common form of RAM, called Dynamic Random Access Memory (DRAM), transistors (acting as switches) and capacitors (acting as storage) come together to form memory cells (each memory cell representing one bit). Memory cells are etched onto a chip in a series of columns and rows (known as arrays); the unique memory address of each cell is where the columns and rows intersect. In order to write data to RAM, the CPU uses the appropriate address line to send the controller the address of the memory cell in an electrical charge.
- 42 Inserting more RAM to improve computer performance is an easy operation for any computer user to undertake.

folders (with their attendant metadata) within the drawers; and the computer file is analogous to a document within the folder.⁴³ However, it is important to point out that this analogy is not perfect, especially given the fact that one computer file may be saved across portions of the file system.

The hard drive is a form of non-volatile random access memory that records data magnetically, with sequential changes in the direction of magnetization representing the 1s and 0s of binary. The hard drive itself consists of components sealed in a protective casing including: stacked, double-sided, disk platters coated with a magnetic medium; a motorized spindle on which the platters sit; a device called a head actuator assembly composed of the read/write heads (tiny electromagnets that sense and record data), control arms that hold the heads, and voice coil actuators that help to direct the read/write heads over the surface of the platters; and an internal circuit board to run the components.⁴⁴ The hard drive connects to the motherboard via an electronic interface device. These electronic interface devices are based on different sets of standards including SCSI (Small Computer System Interface), EIDE (Enhanced Integrated Drive Electronics), and SATA (Serial Advanced Technology Attachment). These devices allow the CPU to interact with, and transfer data between, any type of disc drive including an optical drive or a hard drive. A controller (consisting of a chip and electronic circuits) built into the motherboard directs the transfer of data and instructions to and from the hard drive. From an archival perspective, knowledge of the different types of electronic interface devices is particularly important in dealing with legacy media as it helps archivists understand which connectors will be needed to access different types of hard drives.

Hard drives must be formatted before being used by operating systems and computer applications. Low-level formatting involves physically embedding magnetic markers (1s and 0s) in the magnetic surface of the disk, which divide the disk into tracks, sectors, and clusters to help the disk controller read and write data. Tracks are the concentric circles on the surface of the disk where the information is stored. Tracks are divided into pie-shaped sectors (generally 512 bytes in size) that act as storage areas for data. A cluster is a group of two or more sectors, and it is the smallest amount of disk space that can be used to save a file. Hard drives can also be partitioned into logical units or regions so that a user can gain more control over how data is stored on the device.⁴⁵

High-level formatting sets up a file system on the first sector of the hard

43 Computer scientists define a file as “a named collection of data that exists on a storage medium ... [that] can contain a group of records, a document, a photo, an e-mail message, or a computer program.” June Jamrich Parsons and Dan Oja, *New Perspectives on Computer Concepts 2011* (Boston, MA, 2010), p. 204.

44 Hard drives are characterized by the speed of the platters, measured in revolutions per minute.

45 Partitioning, for example, allows multiple operating systems to be run on one hard drive.

drive. A file system is a database that is needed in order to help the computer's operating system name, store, describe, and retrieve files.⁴⁶ It is the job of the file system to: organize disk sectors into files and directories; keep track of free disk space; and keep track of which files are stored where in a partition. The file directory stores information about the files (including file attributes such as file name, extension, etc.), and maps between the file names and the stored files. There are different types of file systems associated with different operating systems (File Allocation Table [FAT], New Technology File System [NTFS], etc.), and file-naming conventions differ between these file systems. For archivists, a thorough knowledge of file systems is particularly important when copying files from legacy media – ensuring that incompatible characters in file names or differences in the file size accepted by file systems do not become an issue.⁴⁷

While archivists are primarily interested in the user-created files on a hard drive, it is important to note that portions of files and their corresponding metadata can reside in seemingly unlikely places on a computer's hard drive. Two important concepts to understand are the computer's swap space and slack space. Swap space refers to the fact that when a computer is multi-tasking it is not always possible for all data to be held in main memory (RAM). To remedy this issue, the computer moves data back and forth between RAM and the hard drive. When a Windows operating system shuts down, some of this data is saved to the hard drive, either in unallocated space or in something called slack space. Slack space is the remainder space that exists when a file is saved across several clusters. This slack space may contain data that the computer cleared from memory when the user closed a file (RAM slack), or data from a file that was stored there previously (drive slack).

The Black Box – Software

Having looked at the computer and at the record as a physical object, we must now turn our attention to how digital records can be understood as logical objects or “processable units” (objects that can be recognized and processed by programs and software).⁴⁸ This involves looking more deeply into the functions of computer software, as well as how binary data is interpreted and recognized by computer software. Software is a set or sequence of instructions (called programs) that allow the computer to operate. Without software, the computer hardware is inert; it does not know what to do. Software is written in various

46 On a PC running a Microsoft operating system, information about the file system of any internal or attached external drive can be found by looking under Administrative Tools → Computer Management → Storage → Disk Management. This information includes the file system type, capacity, and amount of free storage.

47 Kirschenbaum, Ovenden, and Redwine, *Digital Forensics*, pp. 15–17.

48 Thibodeau, p. 4.

programming languages that can be categorized in relationship to the degree to which they resemble human language or artificial language, and the degree to which they require knowledge of a computer’s architecture. The programming language at the lowest level of abstraction, composed of binary digits or bits, is known as machine language (also known as machine code or object code). Machine language exists as a series of instructions and data that can be directly interpreted by a computer’s CPU. Each type of processor or family of processors has its own set of machine code, called an instruction set. An instruction is typically composed of an operation code (opcode) that specifies the function of the instruction that is to be performed by the processor (add, subtract, compare, store a value in memory, etc.), and the entities on which these operations are to be performed (these operands may include the address or location of the data on which the operation is to be performed). The instruction set is implemented through another low-level programming language known as microcode.

Because machine code exists as a series of 1s and 0s, it is difficult for people to read and write this language. Therefore a second-generation language (the next step in the evolution of computer languages) known as assembly language has been developed. Assembly language takes the sequence of instructions created in patterns of 1s and 0s of machine language and replaces them with a human-readable notation of alphanumeric symbols (mnemonics). As each processor or family of processors has its own set of machine code, each processor or family of processors also has its own assembly language. The translation from assembly language back to machine language so that the computer can execute the instructions takes place through a utility program known as an assembler.

Higher-level languages (such as C, Java, and Python) have a greater degree of abstraction from the fundamental workings of the CPU. Higher-level programming languages are also more portable across computer platforms, and are written in a syntax that is easier for humans to read, write, and maintain; however, high-level languages must be converted into machine code in order to be understood by the computer. Converting the source code to machine code for each high-level language requires the use of a particular computer program known as a compiler. After converting the source code into machine language, the compiler generally saves the results as a separate binary file (also called an executable file).⁴⁹ This executable file exists independently of the source code and, as such, can be run at any time by simply clicking on it. An alternative to compiling a program is to use a program called an interpreter. Interpreters take high-level languages and translate and carry out the instructions in a line-by-line manner.

49 Using open-source software sites like SourceForge.net, executable files can often be downloaded without having to compile the code itself.

Not only is software written in various programming languages, but it also comes in various types, including platform or system software, middleware, and application software. System software (such as BIOS, operating systems, and utility software such as device drivers and disk defragmenters) controls and manages a computer's resources (CPU, RAM, file storage, input/output devices, etc.), and provides a set of services (such as file, process, and memory management) for applications that are running on a computer. Middleware is a software layer that resides between the operating system or networking software, and the computer applications. The job of middleware is to facilitate communication or interoperability between applications. Examples of middleware include mail server software, web servers, and database management systems. Application software is software that is created to perform a specific task for an end use, and it is at the level of the application software that we traditionally think of the digital record as coming into being. Although powerful in its own right, application software relies on the operating system to manage many background tasks, including memory management. Examples of application software used on a PC include database programs, word processing software, and web browsers. The process of writing and running application and system software is made more efficient through the use of a collection of executable files called dynamic-link libraries (DLL). These libraries are part of the operating system and/or application software, and contain reusable code that is written to perform a specific function (spelling and grammar checking, for example); they can be called by other applications or components of the operating system and loaded into main memory and run as needed.⁵⁰ Device drivers, which support the operation of specific devices (such as printers), can be packed as DLLs.

Operating systems (OS) are usually written in assembly language and/or higher-level languages such as C. The OS is usually installed on the hard drive (usually in a primary partition known as the root folder), and portions are pulled into main memory (RAM) as the user interacts with the operating system. Operating systems are designed in layers. The shell of the operating system is a software program that acts as an interface between the user and the operating system, interpreting operating system commands and eliciting operating services from the kernel. The kernel is a layer or component of the operating system that is located closest to the hardware (particularly the CPU and memory), and consists of the essential routines or services performed by the operating system (memory, file, process, and input/output management).⁵¹ These services are requested by other parts of the operating system or by

50 One of the main advantages of DLLs is that they free up memory space.

51 Generally, the kernel is the only layer that is allowed to interact with the hardware. Applications and other programs generally talk to the user interface layer, which in turn talks to the kernel.

applications through a mechanism known as system calls. The kernel is the first part of the operating system that is loaded into memory when the computer is booted up, and it remains in memory during the computer session.

While key functions of the operating system include file, memory, and process management, carrying out these and other functions includes recognizing inputs from devices, transmitting output to devices, and controlling peripherals; creating, deleting, and searching files and file directories; reading and writing files onto secondary storage; allocating space and managing the file system and directory on the hard drive; tracking and scheduling threads or processes (those parts of software that perform a controllable action) to be executed by the CPU; managing memory allocation and use so that multiple processes are executed using available memory; providing a way for applications to interact with the computer’s hardware through the use of application program interfaces (APIs); interfacing with device drivers thus allowing devices (such as printers and floppy disk drives) and the operating system to communicate with each other; providing programs to monitor the performance and security of a computer system; and providing a structure or system (generally command line or a graphical user interface) through which a person can directly interact with the operating system. In a Windows operating system a centralized database called a registry acts as the brains of the operating system. The registry houses hierarchical batches of files that consist of the settings needed by the operating system and by software applications. In effect, the registry documents how a computer system has been configured and used, and includes data of possible interest to archivists including user and password data, as well as information on which programs have been installed and uninstalled, what hardware has been connected to the computer, and what documents have been opened and closed.

Digital objects are recognized and processed by programs and software through the use of various encoding schemes. In particular, software interprets character data from language or writing systems through encoding schemes such as EBCDIC, ASCII, and Unicode, which translate each upper and lower case letter, numerical digit, punctuation mark, symbol, etc. into a unique combination of bits.⁵² Each file is associated or saved with character encoding (sometimes known as a character set), and the choice of character set can be an interaction between the operating system, the application, and the user. The character encoding is the map between the text in a file that exists as binary

52 Extended Binary Coded Decimal Interchange Code (EBCDIC) is an 8-bit, 256-character set developed by International Business Machines (IBM) primarily for mainframe computers. The American Standard Code for Information Interchange (ASCII) is a 7-bit, 128-character set developed by the American Standards Association and widely used with personal computers. Unicode is an international, 32-bit, character-encoding scheme that, because of its much larger character set, can represent any character or symbol in modern and ancient writing scripts.

digits stored in memory and how that text will render when output (on a monitor, printer, etc.). Not only does each file have a character set, it also has rules that specify how the bytes that make up the file are organized, interpreted, and rendered (referred to as a file format). Different computer applications have their own native file formats and other file formats in which they can open and save documents.⁵³ Operating systems and applications determine the format of a file through various methods: through information stored within the file itself, through information stored in the file system, or based on the filename extension.

As it is of particular interest to archivists, it is helpful to reiterate here that files have associated metadata stored in a variety of places within the computer system. The file system contains metadata about the file, including location, size, date last modified, creation date, and owner of the file. The system registry contains metadata about the computer and about the computer's user. And as an instantiation of a file format, the document itself can store additional internal metadata, which can be added whether by device, application, or user. This metadata includes information on who created or saved a document, and when it was created, edited, or printed.⁵⁴

Putting all of this information together, it should be much clearer now how born-digital records are forged in the alliance of the user, the computer hardware, the operating system software, and the application software. This journey starts with a user sitting in front of a computer. By pushing a button to power on a PC, a user sets a process in motion. The BIOS begins by loading and then starting the operating system. The user interacts with the operating system through a user interface – a window that allows the user to engage with the system by clicking, double-clicking, pointing, or typing. While a command line interface provides a text prompt in which a user can enter specific commands to be executed, a graphical user interface allows the user to interact with the computer system through menus and icons. If a user clicks the icon for the word processing software, the registry is queried to find where the program is stored. The program is loaded from the hard drive into main memory (RAM) where it executes, and the application window becomes available to the user. The user can now begin to work on a document.

When a user is working within a word processing application, the software is also creating a temporary work file that is stored in main memory (RAM). When the user clicks an icon to save a word processing document, the operat-

53 Computer files are presented according to predefined structural and organizational principles, referred to as file formats. File formats are typically laid out in a document called a format specification. File formats can be proprietary or open.

54 The new Microsoft Word file format (.docx) is actually a collection of XML files stored in a Zip file. In this file format, metadata extracted from the application and metadata extracted from the document itself are stored in two XML files in a folder called docProps.

ing system steps in, asking the user to assign a file name and to select where the document is to be stored. When a name is assigned to the document, the operating system checks whether the file name is valid. The application software also hands the operating system the job of saving the file from RAM to permanent storage. The operating system modifies the file system to indicate that a file will be stored in a particular location, and checks the file system to see which clusters on the hard disk are free. The operating system sends this information to BIOS, and BIOS in turn retrieves the data that makes up the file from RAM. BIOS also issues instructions to the controller mechanism on the disk drive. The controller mechanism translates the instructions from BIOS into the electrical signals that move the hard drive’s read/write head, thus allowing the data to be recorded on the hard drive. The operating system records the location information in the file system so that the data on the hard drive is not overwritten. The document is born.

Conclusion

As Kirschenbaum so eloquently puts it, computers are “unique in the history of writing technologies in that they present a premeditated material environment built and engineered to propagate an illusion of immateriality.”⁵⁵ Today, design ideals and notions of what the consumer wants and needs are moving us even deeper into the postmodern aesthetic. According to cognitive scientist and design expert Donald Norman, what people want is an “invisible computer” – technology that is “hidden behind the scenes, providing its benefits without pain, anguish, and stress.”⁵⁶ According to sociologist Bart Simon, this postmodern ideal has been made possible through the miniaturization of computers, changes to their physical design (computers are increasingly camouflaged as everyday objects), as well as innovations in liquid crystal display (LCD) and plasma screens (which allow people to maximize their engagement with “life on the screen”).⁵⁷ Archivists must embrace this postmodern aesthetic in order to engage user communities in ways that make the most sense to them. In particular, we need to understand how to design, deploy, and evaluate the computer interfaces through which we provide resources to our user communities.

However, in order to meet the challenge of preserving and making accessible born-digital records, we must continue to nurture another type of relationship with the computer and the computer system. We have to be, as it were, “modern” postmodernists. This involves opening up the computer’s black box and learning how computer hardware and software function in relation to the creation

55 Kirschenbaum, p. 135.

56 Donald A. Norman, *The Invisible Computer* (Cambridge, MA, 1999), p. x.

57 Bart Simon, “Geek Chic: Machine Aesthetics, Digital Gaming, and the Cultural Politics of the Case Mod,” *Games and Culture*, vol. 2, no. 3 (June 2007), pp. 175–93.

and storage of records and associated metadata. It has been shown here that at least a general knowledge of the relationship between computer hardware, operating system, and application software is necessary in order for archivists to embrace sophisticated models of how digital records exist within computer systems. Understanding the levels of abstraction that form a computer system simplifies the inherent complexity of these machines and shows how records exist as physical, logical, and conceptual objects. The acquisition of this knowledge involves pairing ideas from archival science with those from computer architecture, computer engineering, and computer science. Archivists also need to understand how to access and evaluate relevant hardware and software in order to acquire and preserve born-digital records.⁵⁸ This involves pairing ideas from archival science with those from digital forensics, with a particular focus on understanding those components of a computer system in which records and metadata reside (file systems, Windows registry, BIOS, file slack areas, etc.).

While understanding what goes on inside the computer's black box allows archivists to meet the challenge of preserving and making accessible born-digital records, it has another imperative: supporting archivists as they grapple with a new reference and access paradigm for a community of researchers (particularly in the area of textual studies) who are interested not only in accessing the electronic text but also in studying the computer itself as a material artifact of the writing process. These researchers study the computer, in its totality, so that they can understand it as a writing technology and as an integral part of the environment in which writing and the creative process takes place. Similarly, the computer's setup and customization is studied in relation to the notion that the computer can be understood, and analyzed, as an extension of the self. We can more fully engage with this community of researchers if we have what Kirschenbaum et al. call "a curatorial sensitivity toward the uniqueness of individual instances of both hardware and data objects, coupled with an awareness of how the affordances of particular systems, environments, and technologies can all impact the creative process."⁵⁹

In exploring the landscape of computer hardware and software in which records reside, this modern terrain has been reclaimed as an area of interest for the archival profession. It is a landscape that will continue to grow and change

58 A good example of how archivists are accessing and evaluating relevant hardware and software in order to preserve born-digital records can be seen in the work of Dr. Patricia Galloway and her students at the School of Information at the University of Texas – Austin. See Karen Ballinger, Bonnie Cope, Jocelyn Petyak, and Jessica Meyerson, "Digital Archaeology Lab Manual" (2011), <https://pacer.ischool.utexas.edu/handle/2081/23283> (accessed on 13 August 2011).

59 Matthew G. Kirschenbaum, Erika L. Farr, Kari M. Kraus, Naomi Nelson, Catherine Stollar Peters, Gabriela Redwine, et al., "Digital Materiality: Preserving Access to Computers as Complete Environments," *iPRES 2009: Proceedings of the Sixth International Conference on Preservation of Digital Objects* (Oakland, CA, 2009), pp. 105–112.

as computers and computer systems evolve (from binary computing to quantum computing perhaps), and as computing trends come and go (the current trend toward cloud computing, for example, may make it more difficult for archivists to access and study the physical media on which the records reside).⁶⁰ In exploring this landscape, the intent has also been to show that in the digital realm we are not facing as great a loss of the physical and tangible nature of our work as we had once imagined. On the contrary, if the black box is opened for examination, we can find the equivalent of the record, the file folder, and the filing cabinet, even if they exist at the most basic level as material traces in either electronic or magnetic form. Indeed, as anyone who has opened a computer case recently will attest, even the dust is still there!

60 Archivists will need to pay attention to the trend that is seeing a change from magnetic hard drives in favour of solid-state drives using flash storage. The difference in the inner working of solid-state drives has implications for what data can be retrieved from these storage devices. See Graeme B. Bell and Richard Boddington, “Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?” *The Journal of Digital Forensics, Security and Law*, vol. 5, no. 3 (2010), pp. 1–20.