

Ensuring the Longevity of Digital Documents

The digital medium is replacing paper in a dramatic record-keeping revolution. But such documents may be lost unless we act now

by Jeff Rothenberg

The year is 2045, and my grandchildren (as yet unborn) are exploring the attic of my house (as yet unbought). They find a letter dated 1995 and a CD-ROM. The letter says the disk contains a document that provides the key to obtaining my fortune (as yet unearned). My grandchildren are understandably excited, but they have never before seen a CD—except in old movies. Even if they can find a suitable disk drive, how will they run the software necessary to interpret what is on the disk? How can they read my obsolete digital document?

This imaginary scenario reveals some fundamental problems with digital documents. Without the explanatory letter, my grandchildren would have no reason to think the disk in my attic was worth deciphering. The letter possesses the enviable quality of being readable with no machinery, tools or special knowledge beyond that of English. Because digital information can be copied and recopied perfectly, it is often extolled for its supposed longevity. The truth, however, is that because of changing hardware and software, only the letter will be immediately intelligible 50 years from now.

Information technology is revolutionizing our concept of record keeping in an upheaval as great as the introduction of printing, if not of writing itself. The current generation of digital records has unique historical significance. Yet these

documents are far more fragile than paper, placing the chronicle of our entire period in jeopardy.

My concern is not unjustified. There have already been several potential disasters. A 1990 House of Representatives report describes the narrow escape of the 1960 U.S. Census data. The tabulations were originally stored on tapes that became obsolete faster than expected as revised recording formats supplanted existing ones (although most of the information was successfully transferred to newer media). The report notes other close calls as well, involving tapes of the Department of Health and Human Services; files from the National Commission on Marijuana and Drug Abuse, the Public Land Law Review Commission and other agencies; the Combat Area Casualty file containing P.O.W. and M.I.A. records for the Vietnam War; and herbicide information needed to analyze the impact of Agent Orange. Scientific data are in similar jeopardy, as irreplaceable records of numerous experiments conducted by the National Aeronautics and Space Administration and other organizations age into oblivion.

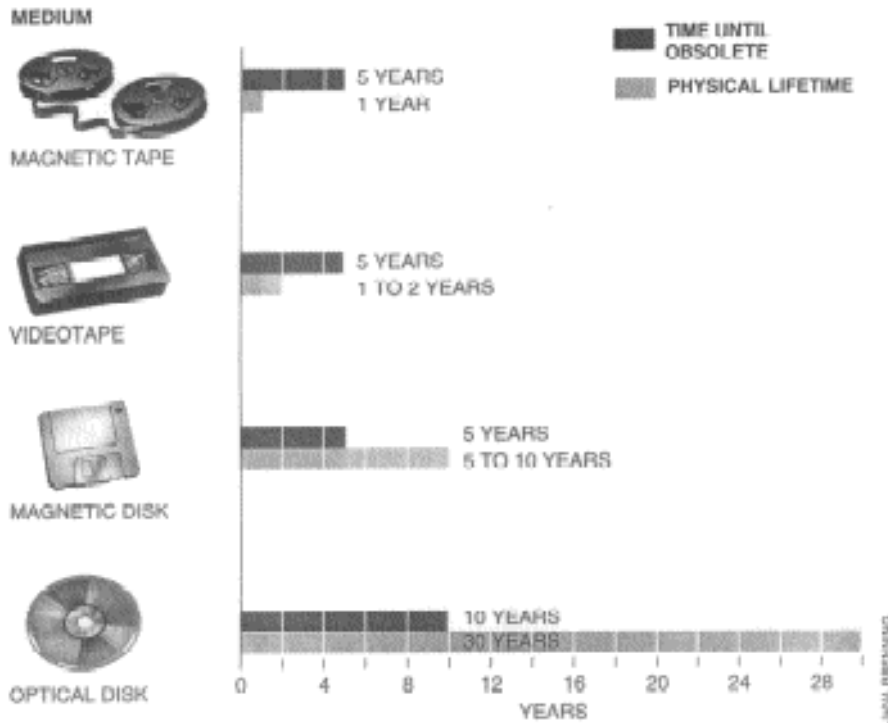
So far the undisputed losses are few. But the significance of many digital documents—those we consider too unimportant to archive—may become apparent only long after they become unreadable. Unfortunately, many of the traditional methods developed for ar-

chiving printed matter are not applicable to electronic files. The content and historical value of thousands of records, databases and personal documents may be irretrievably lost to future generations if we do not take steps to preserve them now.

From Here to Eternity

Although digital information is theoretically invulnerable to the ravages of time, the physical media on which it is stored are far from eternal. If the optical CD in my attic were a magnetic disk, attempting to read it would probably be futile. Stray magnetic fields, oxidation and material decay can easily erase such disks. The contents of most digital media evaporate long before words written on high-quality paper. They often become unusably obsolete even sooner, as media are superseded by new, incompatible formats—how many readers remember eight-inch floppy disks? It is only slightly facetious to say that digital information lasts forever—or five years, whichever comes first.

Yet neither the physical fragility of digital media nor their lemminglike tendency toward obsolescence constitutes the worst of my grandchildren's problems. My progeny must not only extract the content of the disk but must also interpret it correctly. To understand their predicament, we need to examine the nature of digital storage. Digital infor-



EXPECTED LIFETIMES of common digital storage media are estimated conservatively to guarantee that none of the data are lost. (Analog tapes, such as those used for audio recordings, remain playable for many years because they record more robust signals that degrade more gradually.) The estimated time to obsolescence for each medium refers to a particular recording format.

components, we must know the length of a byte.

One way to convey the length is to encode a "key" at the beginning of the bit stream. But this key must itself be represented by a byte of some length. A reader therefore needs another key to understand the first one. Computer scientists call the solution to such a recursive problem a "bootstrap" (from the fanciful image of pulling oneself up by the bootstraps). In this case, a bootstrap must provide some context, which humans can read, that explains how to interpret the digital storage medium. For my grandchildren, the letter accompanying the disk serves this role.

After a bit stream is correctly parsed, we face another recursive problem. A byte can represent a number or an alphabetic character according to a code. To interpret such bytes, therefore, we need to know their coding scheme. But if we try to identify this scheme by inserting a code identifier in the bit stream itself, we will need another code identifier to interpret the first one. Again, a human-readable context must serve as a bootstrap.

Even more problematic, bit streams may also contain complex cross-referencing information. The stream is often stored as a collection, or file, of bits that contains logically related but physi-

cally separate elements. These elements are linked to one another by internal references, which consist of pointers to other elements or of patterns to be matched. (Printed documents exhibit similar schemes, in which page numbers serve as pointers.)

Interpreting a Bit Stream

Suppose my grandchildren manage to read the bit stream from the CDROM. Only then will they face their real challenge: interpreting the information embedded in the bit stream. Most files contain information that is meaningful solely to the software that created them. Word-processing files embed format instructions describing typography, layout and structure (titles, chapters and so on). Spreadsheet files embed formulas relating their cells. So-called hypermedia files contain information identifying and linking text, graphics, sound and temporal data.

For convenience, we call such embedded information—and all other aspects of a bit stream's representation, including byte length, character code and structure—the encoding of a document file. These files are essentially programs: instructions and data that can be interpreted only by appropriate software. A file is not a document in its own right—

it merely describes a document that comes into existence when the file is interpreted by the program that produced it. Without this program (or equivalent software), the document is a cryptic hostage of its own encoding.

Trial-and-error might decode the intended text if the document is a simple sequence of characters. But if it is complex, such a brute-force approach is unlikely to succeed. The meaning of a file is not inherent in the bits themselves, any more than the meaning of this sentence is inherent in its words. To understand any document, we must know what its content signifies in the language of its intended reader. Unfortunately, the intended reader of a document file is a program. Documents such as multimedia presentations are impossible to read without appropriate software: unlike printed words, they cannot just be "held up to the light."

Is it necessary to run the specific program that created a document? In some cases, similar software may at least partially be able to interpret the file. Still, it is naive to think that the encoding of any document—however natural it seems to us—will remain readable by future software for very long. Information technology continually creates new schemes, which often abandon their predecessors instead of subsuming them.

A good example of this phenomenon occurs in word processing. Most such programs allow writers to save their work as simple text, using the current seven-bit American Standard Code for Information Interchange (or ASCII). Such text would be relatively easy to decode in the future if seven-bit ASCII remains the text standard of choice. Yet ASCII is by no means the only popular text standard, and there are proposals to extend it to a 16-bit code (to encompass non-English alphabets). Future readers may therefore not be able to guess the correct text standard. To complicate matters, authors rarely save their work as pure text. As Avra Michelson, then at the National Archives, and I pointed out in 1992, authors often format digital documents quite early in the writing process and add figures and footnotes to provide more readable and complete drafts.

If "reading" a document means simply extracting its content—without its original form—then we may not need to run the original software. But content can be lost in subtle ways. Translating word-processing formats, for instance, often displaces or eliminates headings, captions or footnotes. Is this merely a loss of structure, or does it impinge on content? If we transform a spreadsheet into a table, deleting the formulas that

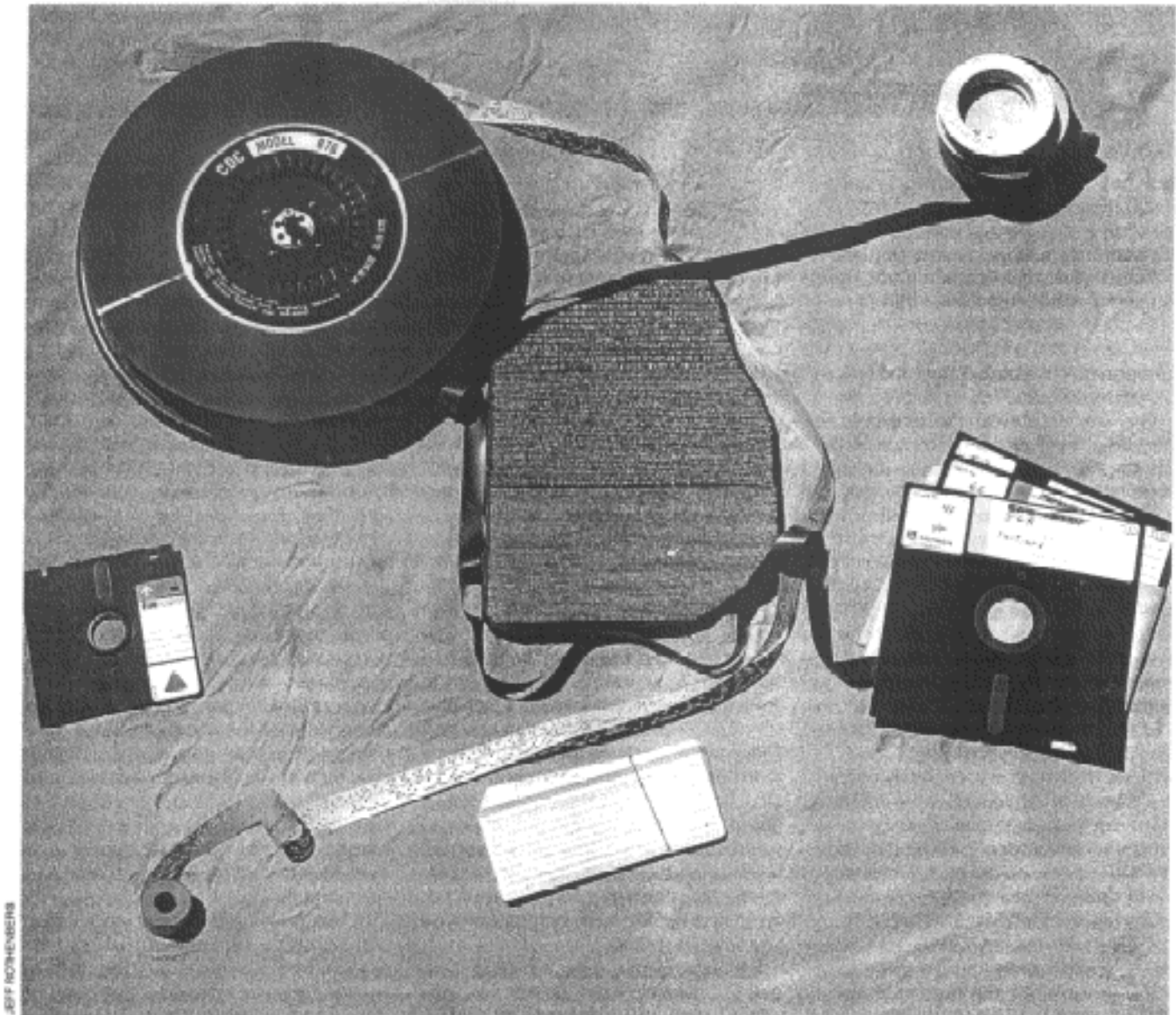
ation can be saved on any medium that able to represent the binary digits (bits") 0 and 1. We will call an intended, meaningful sequence of bits, with no intervening spaces, punctuation or formatting, a bit stream.

Retrieving a bit stream requires a hardware device, such as a disk drive, and special circuitry for reading the physical representation of the bits from the medium. Accessing the device from a given computer also requires a "driver" program. After the bit stream is retrieved, it must still be interpreted. This task is not straightforward, because a given bit stream can represent almost anything—from a sequence of integers

to an array of dots in a pointillist-style image. Furthermore, interpreting a bit stream depends on understanding its implicit structure, which cannot explicitly be represented in the stream. A bit stream that represents a sequence of alphabet

JEFF ROTHENBERG is a senior computer scientist in the social policy department of the RAND corporation in Santa Monica, Calif. He received a master's degree in computer science from the University of Wisconsin in 1969 and then spent the next four years working toward a doctorate in artificial intelligence. His research has included work in modeling theory, investigations into the effects of information technology on humanities research, and numerous studies involving information technology policy issues. His passions include classical music, traveling, photography and sailing.

ic characters may consist of fixed-length chunks ("bytes"), each representing a code for a single character. For instance, in one current scheme, the eight bits 01110001 stand for the letter q. To extract the bytes from the bit stream, thereby "parsing" the stream into its



OBSOLESCENCE plagues digital media. Those shown have already failed to remain readable for one hundredth the time that the Rosetta Stone has. The classical Greek script in the stone, which was found in 1799 in Egypt by a French military

demolition squad, made hieroglyphics and demotic Egyptian comprehensible. Besides being legible after 22 centuries, the Rosetta Stone (a replica here) owes its preservation to the visual impact of its content—an attribute absent in digital media.

relate the table's entries to one another, have we affected content? Suppose the CD in my attic contains a treasure map depicted by the visual patterns of word and line spacings in my original digital version of this article. Because these patterns are artifacts of the formatting algorithms of my software, they will be visible only when the digital version is viewed using my original program. If we need to view a complex document as its author viewed it, we have little choice but to run the software that generated it.

What chance will my grandchildren have of finding that software 50 years from now? If I include a copy of the program on the CD, they must still find the operating system software that allows the program to run on some computer. Storing a copy of the operating system on the CD may help, but the computer hardware required to run it will have long since become obsolete. What kind of digital Rosetta Stone can I leave to provide the key to understanding the contents of my disk?

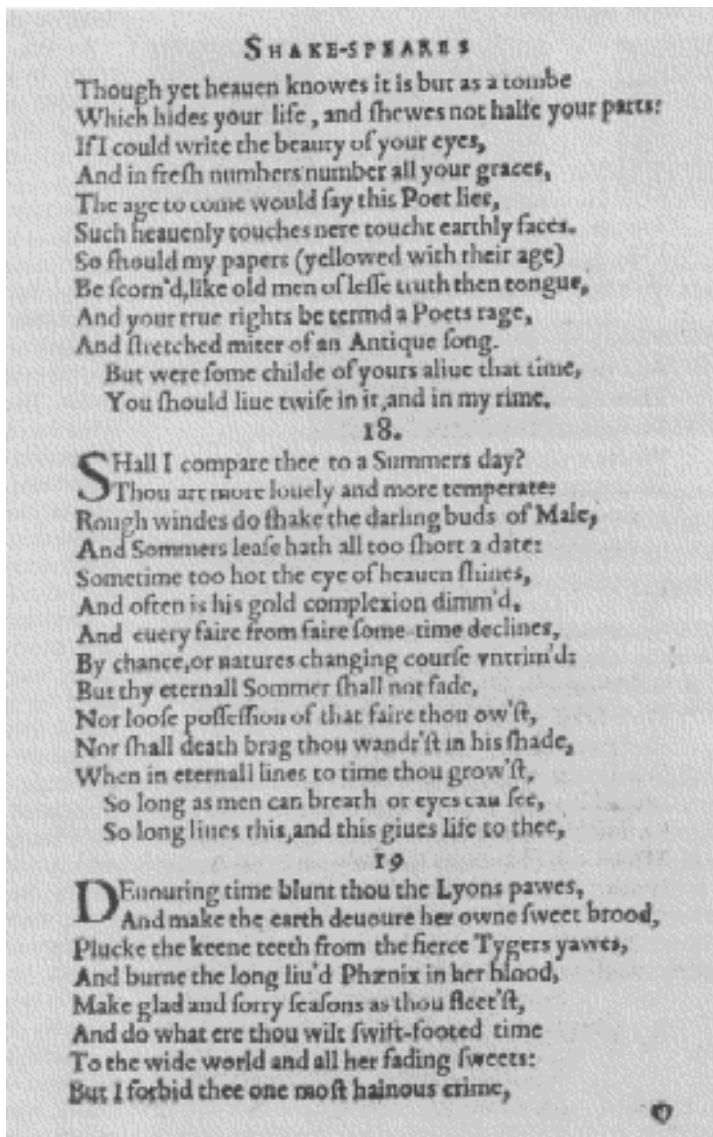
Migrating Bits

To prevent digital documents from being lost, we must first preserve their bit streams. That means copying the bits onto new forms of media to ensure their accessibility. The approach is analogous to preserving text, which must be transcribed periodically. Both activities require ongoing effort: future access depends on an unbroken chain of such migrations frequent enough to prevent media from becoming physically unreadable or obsolete before they are copied. A single break in this chain renders digital information inaccessible, short of heroic effort. Given the current lack of permanence of media and the rate at which their forms evolve, migration may need to be as frequent as once every few years. Conservative estimates suggest that data on digital magnetic tape should be copied

Once a year to guarantee that none of the information is lost. (Analog tapes may remain playable for many years because they record more robust signals that degrade more gradually.)

In the long run, we might be able to develop long-lived storage media, which would make migration less urgent. At the moment, media with increased longevity are not on the horizon. Nevertheless, the cost of migration may eventually force the development of such products, overriding our appetite for improved performance.

An ancient text can be preserved either by translating it into a modern language or by copying it in its original dialect. Translation is attractive because it avoids the need to retain knowledge



SHAKESPEARES first printed edition of sonnet 18 (1609) exemplifies the longevity of the printed page: the words are legible after almost four centuries (the final couplet is especially relevant to preserving documents). But digital media can become unreadable within a decade.

Of the text's original language, yet few scholars would praise their predecessors for taking this approach. Not only does translation lose information, it also makes it impossible to determine what information has been lost, because the original is discarded. (In extreme cases, translation can completely undermine content: imagine blindly translating both languages in a bilingual dictionary into a third language.) Conversely, copying text in its original language (saving the bit stream) guarantees that nothing will be lost. Of course, this approach assumes that knowledge of the original language is retained.

Archivists have identified two analogous strategies for preserving digital documents. The first is to translate them into standard forms that are independent of any computer system. The second approach is to extend the longevity of computer systems and their original software to keep documents readable. Unfortunately, both strategies have serious shortcomings.

On the surface, it appears preferable to translate digital documents into standard forms that would remain readable in the future, obviating the need to run obsolete

software. Proponents of this approach offer the relational database (introduced in the 1970s by E. F. Codd, now at Codd & Date, Inc., in San Jose, Calif.) as a paradigmatic example. Such a database consists of tables representing relations among entities. A database of employees might contain a table having columns for employee names and their departments. A second table in the database might have department names in its first column, department sizes-in its second column and the name of the department head in a third. The relational model defines a set of formal operations that make it possible to combine the relations in these tables—for example, to find the name of an employee's department head.

the software and thereby read the document. But information science cannot yet describe the behavior of software in sufficient depth for this approach to work, nor is it likely to be able to do so in the near future. To replicate the behavior of a program, there is currently little choice but to run it.

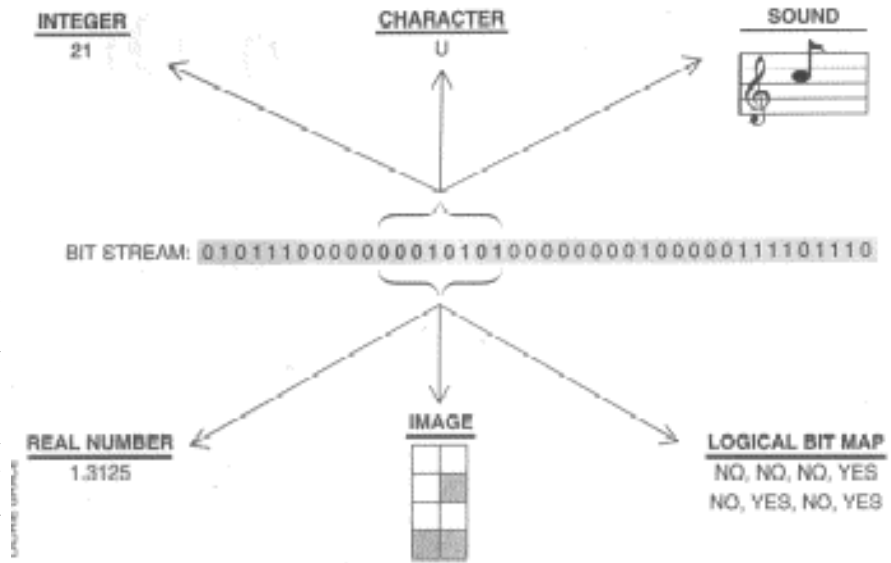
For this reason, we must save the programs that generate our digital documents, as well as all the system software required to run those programs. Although this task is monumental, it is theoretically feasible. Authors often include an appropriate application program and operating system to help recipients read a digital document. Some applications and system software may remain ubiquitous, so that authors would need only to refer readers to those programs. Free, public-domain software is already widely available on the Internet. Moreover, when proprietary programs become obsolete, their copyright restrictions may expire, making them available to future users.

How can we provide the hardware to run antiquated systems and application software? A number of specialized museums and "retro-computing" clubs are attempting to maintain computers in working condition after they become obsolete. Despite a certain undeniable charm born of its technological bravado, this method is ultimately futile. The cost of repairing or replacing worn out components (and retaining the expertise to do so) must inevitably outweigh the demand for any outmoded computer.

Fortunately, software engineers can write programs called emulators, which mimic the behavior of hardware. Assuming that computers will become far more powerful than they are today, they should be able to emulate obsolete systems on demand. The main drawback of emulation is that it requires detailed specifications for the outdated hardware. To be readable for posterity, these specifications must be saved in a digital form independent of any particular software, to prevent having to emulate one system to read the specifications needed to emulate another.

Saving Bits of History

If digital documents and their programs are to be saved, their migration must not modify their bit streams, because programs and their files can be corrupted by the slightest change. If such changes are unavoidable, they must be reversible without loss. Moreover, one must record enough detail about each transformation to allow reconstruction of the original encoding of the bit stream. Although bit streams



INTERPRETING A BIT STREAM correctly is impossible without contextual information. This eight-bit sequence can be interpreted in at least six different ways.

can be designed to be immune to any expected change, future migration may introduce unexpected alterations. For example, aggressive data compression may convert a bit stream into an approximation of itself, precluding a precise reconstruction of the original. Similarly, encryption makes it impossible to recover an original bit stream without the decryption key.

Ideally, bit streams should be sealed in virtual envelopes: the contents would be preserved verbatim, and contextual information associated with each envelope would describe those contents and their transformation history. This information must itself be stored digitally (to ensure its survival), but it must be encoded in a form that humans can read more simply than they can the bit stream itself, so that it can serve as a bootstrap. Therefore, we must adopt bootstrap standards for encoding con

textual information; a simple, text-only standard should suffice. Whenever a bit stream is copied to new media, its associated context may be translated into an updated bootstrap standard. (Irreversible translation would be acceptable here, because only the semantic content of the original context need be retained.) These standards can also be used to encode the hardware specifications needed to construct emulators.

Where does this leave my grandchildren? If they are fortunate, their CD may still be readable by some existing disk drive, or they may be resourceful enough to construct one, using information in my letter. If I include all the relevant software on the disk, along with complete, easily decoded specifications for the required hardware, they should be able to generate an emulator to run the original software that will display my document. I wish them luck.

FURTHER READING

TEXT AND TECHNOLOGY: READING AND WRITING IN THE ELECTRONIC AGE. Jay David Bolter in *Library Resources and Technical Services*, Vol. 31, NO. 1, pages 1223; January-March 1987.

TAKING A BYTE OUT OF HISTORY: THE ARCHIVAL PRESERVATION OF FEDERAL COMPUTER RECORDS Report 101-978 of the U S House of Representatives Committee on Government Operations, November 6, 1990

ARCHIVAL MANAGEMENT OF ELECTRONIC RECORDS Edited by David Bearman Archives and Museum Informatics, Pittsburgh, 1991

UNDERSTANDING ELECTRONIC INCUNABULA: A FRAMEWORK FOR RESEARCH ON

ELECTRONIC RECORDS: Margaret Hedstrom in *American Archivist*, Vol. 54, No. 3, pages 334-354; Summer 1991

ARCHIVAL THEORY AND INFORMATION TECHNOLOGIES: THE IMPACT OF INFORMATION TECHNOLOGIES ON ARCHIVAL PRINCIPLES AND PRACTICES. Charles M Dollar Edited by Oddo Bucci. Information and Documentation Series No. 1, University of Macerata, Italy, 1992.

SCHOLARLY COMMUNICATION AND INFORMATION TECHNOLOGY: EXPLORING THE IMPACT OF CHANGES IN THE RESEARCH PROCESS ON ARCHIVES. Avra Michelson and Jeff Rothenberg in *American Archivist*, Vol. 55, No. 2, pages 236-315; Spring 1992