Communications of the Association for Information Systems

Volume 3 Article 2

January 2000

# The Digital Rosetta Stone: A Model for Maintaining Long-term Access to Static Digital Documents

Alan R. Heminger
*Air Force Institute of Technology*, alan.heminger@afit.af.mil

Steven Robertson
*Air Force Institute of Technology*, steven.robertson@afit.af.mil

Recommended Citation

Heminger, Alan R. and Robertson, Steven (2000) "The Digital Rosetta Stone: A Model for Maintaining Long-term Access to Static Digital Documents," *Communications of the Association for Information Systems*: Vol. 3 , Article 2.
DOI: 10.17705/1CAIS.00302
Available at: https://aisel.aisnet.org/cais/vol3/iss1/2

Communications of the Association for Information Systems

**Volume 3, Article 2**
**January 2000**

# THE DIGITAL ROSETTA STONE: A MODEL FOR MAINTAINING LONG-TERM ACCESS TO STATIC DIGITAL DOCUMENTS

Alan R. Heminger
Air Force Institute of Technology

Steven B. Robertson
Captain
United States Air Force

alan.heminger@afit.af.mil

RESEARCH

# THE DIGITAL ROSETTA STONE: A MODEL FOR MAINTAINING LONG-TERM ACCESS TO STATIC DIGITAL DOCUMENTS

Alan R. Heminger
Air Force Institute of Technology
alan.heminger@afit.af.mil

Steven B. Robertson
Captain
United States Air Force

alan.heminger@afit.af.mil

## ABSTRACT

In the past several decades, and at an increasing pace, many records that used to be stored on paper have been stored digitally on computer information systems, instead. As older technologies are replaced by newer generations of hardware and software, new schemes for storing and coding the data are introduced. Because of the rapid evolution of technology, future digital systems may not be able to read and/or interpret the digital records made and stored on these older systems, even if those records are still in good condition. We are losing the knowledge of how the old systems stored and coded information. Increasingly, therefore, when we attempt to access and recover those aging documents, we will find that we no longer have the necessary information to do that.

This paper addresses the problem of maintaining long-term access to digital documents and provides a methodology for overcoming access difficulties due to technological obsolescence. We created a model, called the Digital Rosetta Stone, that provides a methodology for maintaining long-term access to digital documents. The underlying principle of the model is that knowledge preserved about different storage devices and file formats can be used to recover data from obsolete media and to reconstruct the digital documents. We describe three processes that are necessary for maintaining long-term access to digital documents in their native formats--knowledge preservation, data recovery, and document reconstruction.

**Keywords:** digital documents, document access, storage technologies, document recovery, long term access

## I. INTRODUCTION

Virtually all large organization need to retain and, on occasion, refer to various stored documents. Until recently, documents were generally paper or microfilm based. However, modern data storage methods include digital storage. More and more organizations store more and more of their organizational information digitally. Furthermore, rapid evolution of computer technology makes available new generations of systems, which are more capable and more cost effective, and thus, are quickly adopted. . As the older technologies are replaced by newer generations of hardware and software, new schemes for storing and coding the data are introduced. Over time, we are losing our knowledge of how the previous generations of systems stored and accessed documents. With the rapid evolution of storage technologies, future digital systems may not be able to read and/or interpret the digital records made by older systems, even if those records are still in good condition [OASD, 1995].

Within many organizations today, digital documents that are official records must be categorized and managed in accordance with approved records schedules. These records must be retained and accessible throughout their life cycle in accordance with the same laws and standards that govern paper records. In the case of government documents, for example, the law dictates that an official Government record must be classified into one of 26 retention periods set forth by the Archivist of the United States. These retention periods range from 30 days to Permanent storage and include time periods of 30 years, 50 years, and 75 years.

Digital documents that require long retention periods face accessibility problems because of the technology obsolescence of hardware and software. As time goes on, we can expect the problems to become worse, as more and more documents are stored digitally. The National Research Council [1995] best describes this problem in the following statement:

> "The fact that most electronic hardware is expected to function for no more than 10 to 20 years raises very serious problems for long-term (more than 20 years) archival preservation. Even if the operating systems and documentation problems somehow are dealt with, what is the archivist to do when the machine manufacturer declares the hardware obsolete or simply goes out of business? Will there be an IBM or Sony in the year 2200? If they still exist, will they maintain a 1980-1990 vintage machine? Moreover, it must be realized that no archival organization can hope realistically to maintain such hardware itself. Integrated circuits, thin film heads, and laser diodes cannot be repaired today, nor can they be readily fabricated, except in multimillion-dollar factories."

As digital technology continues to evolve rapidly, superseded technologies are discarded quickly and new technologies are embraced in the hopes of gaining improved efficiency, effectiveness, or a competitive advantage. It is crucial that, in the haste to adopt newer and generally better technologies, we don’t lose the ability to access historical digital documents.

The purpose of this paper is to address this issue by developing a conceptual model, which may help us to develop the means to assure ongoing access to the ever-increasing repository of digitally stored information. Rothenberg [1995] used the term digital Rosetta Stone to indicate a means for those in the future to be able to read documents stored on what will then be antiquated information systems technology. In his article in *Scientific American*, he asked what digital Rosetta Stone he should leave with a disk so that generations yet unborn could read it in years to come. We take his suggestion, but recommend that the development of a digital Rosetta Stone not be the individual responsibility of each person who leaves a digital document to posterity. Instead, we recommend that a Digital Rosetta Stone Office be created that assumes responsibility for acquiring and maintaining the necessary knowledge to access documents from obsolete systems. While such an undertaking would probably not be able to capture all the knowledge necessary to retrieve all of the types of digital documents that have been and will be created, it should be able to provide a means to retrieve many of the static documents which have been and are being stored, and which are at risk for becoming irretrievably lost as generations of computer equipment become obsolete and their workings lost to memory. We define static documents as those documents that are basically cognates of paper documents. Such documents would not change their content as a result, say, of database queries by the “reader”.

We will concentrate on maintaining access to digital documents in their native formats without converting them to emerging digital format standards. The resource and financial burdens of converting an entire archival collection every 10 to 20 years is “likely to be out of the question except for relatively small collections that have great historical importance, sustain heavy use, or require rapid access [National Research Council, 1995].” This study is largely exploratory and prescriptive in nature because of the relative newness of this subject area.

As information systems are upgraded, the ability to view digital documents in superseded formats becomes a problem as the hardware and software systems needed to access them become obsolescent. Most digital documents contain information that is only meaningful to the software and hardware systems that were used to create, edit, and access them. Therefore, because of evolving standards for storing digital documents, organizations that archive such documents must develop a method that will allow them to maintain continual access to those documents even after they are stored in what have become superceded formats.

Long-term access to digital documents is also affected by the fact that current, generally available, digital storage media are not stable. That is, over time the information stored on them degrades, and eventually becomes unrecoverable. This is a serious issue that requires a long-term solution.

However, that subject is complex in its own right, and is not within the scope of this paper. Instead, we focus on the information that is still recoverable, but because of the technological changes, is at risk of no longer being readable.

## II.CURRENT STRATEGIES FOR PRESERVING DIGITAL DOCUMENTS

A number of strategies for preserving digital documents are discussed in the literature. Dollar [9] suggested that customers should demand that vendors provide cost-effective migration paths to advancing hardware and software systems. Many vendors do provide a limited form of this capability in that an advanced version of their hardware or software system provides the ability to migrate a customer's operations from the old system to the new one. However, this type of conversion is generally limited to the previous generation of the hardware or software system and therefore is a short-term fix that must be repeated with each successive system upgrade. In addition, the translation of a digital document into successive short-term standards over its life cycle may result in the loss of the document's original content. Without the original document and the original software to interpret the document accurately, the format and content of the document may be compromised and the original meaning lost.

Dollar [1992] also suggested promoting a "trend toward non-proprietary standardized open systems environments, which are designed to overcome compatibility between computer systems and applications and are reflected in international standards". While these open system standards would make digital documents accessible through any software system that conforms to the standards, there is still the problem that even the open system standards will change as information systems technologies advance. Thus, over time, as hardware and software systems evolve, it will still be necessary to either migrate digital documents to an updated standardized format or to provide some other method to maintain continual access to these documents.

Rothenberg [1995] discussed the idea of maintaining long-term access to the information contained within digital documents by extending the life of the original computer hardware and software systems on which the digital documents were created. These life-cycle extensions involve the operation and maintenance of antiquated hardware systems and the archiving of the software needed to access digital documents in their native formats.

While maintaining a depository of antiquated hardware might be achievable in principle, Rothenberg believed that it is also plagued with problems. The main drawbacks being the cost of operating multiple information systems and the difficulty in acquiring antiquated hardware system components [National Research Council, 1995]. These problems make it unrealistic to expect that any organization could effectively and efficiently maintain multiple, aging information technologies in order to maintain access to superseded digital documents.

To overcome the problems associated with maintaining aging hardware, Rothenberg [1995] suggested the creation and use of system emulators that can imitate the behavior of antiquated hardware systems. This method would allow the operation of superseded software on advanced systems as a way to view digital documents in their native formats. However, to emulate an antiquated information system this method requires exhaustive specifications on the original system's hardware. Therefore, this method may require extensive participation by hardware manufacturers. Many manufacturers may be reluctant to supply all of the specifications to software developers because some of the technology may still be in use in the advanced systems they develop.

## III. WHICH STRATEGY TO USE?

None of the strategies discussed in Section II is entirely satisfactory by itself. Therefore, as information systems and their operating environments continue to evolve it may be necessary to use some combination of one or more of these strategies to maintain access to digital documents in superseded formats. The strategies chosen will need to evolve from organizational requirements and conform to the limits of its financial, physical, and human resources [Peterson, 1991].

Because a long-term strategic plan may call for a mix of the methods, it is conceivable that no existing organization can afford the resources necessary to carry out such a tremendous task. Therefore, it may be necessary to establish organizations or processing centers that specialize in maintaining long-term access to digital documents [National Research Council, 1995]. To recapture the information in the myriad digital documents that will be an increasingly large proportion of our information storage may require something comparable to the Rosetta Stone that opened up the writings of ancient Egypt to scholars of today.

## IV. THE ROSETTA STONE

At some point during the fourth century, all knowledge of ancient Egyptian scripts was lost, leaving no method available to decipher the language of hieroglyphics which had been richly preserved on ancient Egyptian monuments, stone tablets, and sheets of papyrus. Fortunately, while on an expedition to Egypt in 1799, Napoleon’s army discovered an artifact that became known as the Rosetta Stone. This stone contained the inscription of a decree issued in 196 BC by Ptolemy V Epiphanes. The decree was repeated three times in two languages, Greek and Egyptian, with the Egyptian version appearing twice, once in hieroglyphics and once in demotic, a cursive form of the hieroglyphic script. Fortunately, there is an abundance of information on ancient Greek dialects and therefore, the stone’s Greek version of the decree contained the key to decipher the meaning of the ancient Egyptian texts. Today, because of the Rosetta Stone, we can interpret many ancient Egyptian texts and inscriptions.

## V. A DIGITAL ROSETTA STONE (DRS)

Rothenberg [1995] asks the question, when considering access to a file that may be generations old, “What kind of digital Rosetta stone can I leave to provide the key to understanding the contents of my disk?" [27] Rothenberg stated that if the behavior of an information system could be sufficiently described, then future generations could re-create that behavior and reproduce digital documents without the need for the original systems. However, he also said that currently information science cannot sufficiently describe this type of behavior in a way that will allow this strategy to succeed, at least not in all cases. Nonetheless, with an increasing amount of our information at risk for becoming unreadable for technological reasons, and recognizing that not all problems can be successfully solved at this time, it is important to begin now to capture what we can that will help us improve the likelihood of retrieving our stored digital documents in the years ahead.

We draw on the strategies discussed above and add others to create a model for maintaining long-term access to digital documents. We call this model the Digital Rosetta Stone (DRS) because, as Rothenberg suggested, it offers a way for those in the future to be able to gain access to the information stored in the digital documents that we have stored, and will continue to store, in increasing numbers. The DRS would contain multiple levels of knowledge about specifications and processes by which information is stored on various types of storage media. It would also contain archives of knowledge about how to interpret that information meaningfully so that the original meaning can be recovered. Ideally, the DRS would become a clearinghouse for knowledge about outdated storage techniques, and outdated hardware and software.

The processes and metadata maintained by the DRS will catalogue the many different aspects of digital technologies. In digital equipment, each component is dependent upon other components of the digital systems to perform a specific task. The process of viewing a file created by a word processor can demonstrate a simplified example of this interdependence. The file must be interpreted by the application program that is dependent upon the operating system which is further dependent upon the system's hardware. Each layer of digital technology involved in this process contributes some form of information necessary to view the digital document.

## VI. DRS COMPONENTS

Unfortunately, creating a DRS is not as simple as the creation of the original Rosetta Stone that held the key to Egyptian hieroglyphics. Instead, a DRS is composed of three major processes that are necessary to preserve and access our digital history:

- knowledge preservation,
- data recovery, and
- document reconstruction.

The knowledge preservation process supports the data recovery and document reconstruction processes.

### KNOWLEDGE PRESERVATION

Knowledge preservation is the process of gathering and preserving the knowledge needed to recover digital data from superseded media and to reconstruct digital documents in their original formats. In a DRS, the preservation of knowledge of hardware and software standards and usage will be maintained in a metaknowledge archive. Metaknowledge is the knowledge or awareness of facts, standards, heuristics, and rules, and the context in which they are used and manipulated. The capture, storage, and availability of this information will provide the metaknowledge necessary to aid document recovery personnel. These repositories will contain the names and descriptions of the data items and processes necessary to recover technologically obsolescent digital documents [Martin, 1989]. We call this collection of information the metaknowledge archive (MKA). The MKA is the foundation upon which the DRS is dependent and it must preserve knowledge extensively in two key areas:

1. The media storage techniques and file formats necessary to recover the bitstream of the documents.
2. The information about the application software necessary to recover the document from the bitstream.

The knowledge of media storage techniques and file formats is a collection of the way data are defined and stored on specific media. Therefore, it is necessary to maintain a record of the methods in which bit patterns are used to represent data on storage devices and how they are arranged within that medium. The knowledge of the location and meaning of these bit patterns will be necessary to recover data if equipment to access a storage medium is not available or no longer exists. This is not to say that all specifications for storage devices must be accurately preserved so engineers can manufacture them in the future. Instead, it requires only that the techniques by which the bit patterns are stored and accessed on the media needs be preserved.

Just as the knowledge of the techniques used to store data on a digital media must be preserved, so we must maintain information on the application software. We must know how the program manipulated and displayed the information. Software applications that create digital documents use data located in specific positions and predefined character sequences to define the digital document's meaning within the application, as well as determine its appearance. Interpretation software is necessary to view a digital document whether it is simply stored in an ASCII text format or in a complex database format, with many complex relationships among the stored data. Software products, commercial-off-the-shelf (COTS) and non-COTS, store and manipulate digital data using a variety of techniques. Therefore, every data file is dependent upon the logic stored within some form of software to properly interpret and display the data file's contents. Although the logical manipulations within software can be exceedingly complex, a simple example illustrates the problem.

*Example:* Consider the simple process of displaying or printing a word processing file accurately. Character sequences embedded within the digital document inform the interpretation software how the document's data is to be interpreted. For example, to bold a section of text using the Hypertext Markup Language (HTML), all characters following the character sequence "<B>" are bolded until the character sequence "</B>" is encountered. Any software capable of interpreting an HTML document must recognize these character sequences and all other format character sequences that are characteristic of HTML documents.

Likewise, any software capable of interpreting a digital document must recognize the formatting character sequences unique to the application that was used to create that digital document.

## DATA RECOVERY

Data recovery is the process of extracting digital data from an obsolete medium and migrating it to a medium that is accessible to current information systems. The recovery will, of course, depend on the cost effectiveness of recovering the data. That is, if the need for the knowledge in the digital document(s) is greater than the cost of recovery, then the cost of the recovery method(s) may be justified.

## DOCUMENT RECONSTRUCTION

Document reconstruction is the process of interpreting digital documents from their original data files by using file format information gathered during the knowledge preservation process. Interpreting digital documents by describing how the original software interpreted the documents is a strategy that was suggested by Michelson and Rothenberg [1992]. While this approach does not exhaustively include all the ways that software can interpret and manipulate stored documents, it does provide a means for addressing the way that software interprets many documents that are electronic versions of what had been stored previously as paper documents. The file format information describes the formatting information used by specific software applications. In other words it is a template that can be used to describe the way data is formatted and displayed by word processing, graphics, and other applications that create digital documents. This does not mean that the algorithms used to produce the documents are preserved so programmers can replicate them in the future. Instead, it means that the bit or character sequences and other formatting information are preserved as a template for document interpreters to use to reconstruct and view documents in their original forms. When the reconstruction process is complete the document should appear in its original form. As in the data recovery process, the methods used during document reconstruction are dependent upon the cost effectiveness of reconstructing the document.

## VI. KNOWLEDGE PRESERVATION

The metaknowledge archive (MKA) is the foundation upon which the DRS is built. It contains hardware and software information, which can be used to extract, and display data in the form prescribed by the information systems used to create digital documents. To ensure the success of the DRS the metaknowledge archive must collect and preserve media storage techniques so engineers can extract the bitstreams from the many different types of media. Likewise, it must also collect and preserve digital document manipulating and formatting information for the different types of software that were in use. While it may not be possible to completely describe all of the manipulations possible with the original software, it should be possible to describe enough of the capabilities sufficiently to allow interpretation of many documents, particularly those which are electronic cognates of paper documents.

However, media storage and document formatting information may not be sufficient for document recovery. A digital document's interpretation also depends on the hardware and systems software that were used to create it. It is the heart of the technological obsolescence problem that newer generations of computers are most often not compatible with the generations of computers that came before. Therefore, it will be necessary to use the capabilities of the newer computers to recover the bitstreams from obsolete media and then to use the new computers along with the knowledge of the original hardware and software in the MKA to devise methods to recover and display the documents created previously.

### MEDIA METADATA

Media metadata is probably the easiest type of data to gather for the DRS because the standards for most storage media are rigidly defined before a medium is brought to market. For example, ISO9660 is the standard that specifies how data are stored on a CD-ROM. This standard defines the volume structures, file structures, and all other attributes associated with a CD-ROM. This type of data must be gathered for each type of media to be included in the metaknowledge archive.

When trying to recover data, recovery personnel must know where to look in order to find it. Media storage geometry defines where on a medium data are stored. For data recovery personnel to find the data they must know the geometric shape of the data's path and the locations of those paths. For example, on a CD-ROM data are stored on a spiraling track with adjacent tracks 1.6 micrometers apart for a track density of 16,000 tracks per inch [Norton et al.,1995]. Furthermore, the tracks are divided into sectors containing 2048 bytes of data and each sector has an address that is used during the file allocation. This type of geometric storage information must be collected for each type of medium.

Once the medium's storage geometry is identified, data recovery personnel must know how data are physically recorded on a medium so that a device can be engineered to read the digital patterns. For example, early storage media stored data as a series of holes punched into paper tape or punched cards. Hard and floppy disks store data as a series of magnetic patterns stored on a layer of magnetic particles. More recent optical technologies, such as the CD-ROM, store data as a series of lands and pits (0.12 micrometers deep and 0.6 micrometers in diameter) burned into a plastic platter. Many other storage methods have been used, are in use, and will be used in the future. Knowing these storage methods tells data recovery personnel what to look for to identify the digital data stored on the medium.

After data recovery personnel have identified where the data are stored, and the data storage method, they must determine how the data are encoded. Encoding techniques define how the data's bit patterns are stored on the media. The encoding information will be used to decode the data and restore the data bit stream to its original form. Encoding schemes may be fairly simple with one setting identifying a 0 bit and another setting defining a 1 bit. Or encoding schemes may implement coding algorithms to encrypt and compress recurring bit patterns. Two popular encoding schemes used today are multiple frequency modulation (MFM) and run length limited (RLL). Multiple frequency modulation is a method of encoding analog signals into magnetic pulses or bits. Run length limited is another method of encoding data into magnetic pulses but its encoding scheme allows 50 percent more data to be stored on a disk than MFM.

During the next step it is necessary to determine the file allocation method used on a media. File allocation is how storage space is assigned to files so that storage space is utilized effectively and files can be accessed [Settani, 1995]. Once data recovery personnel can locate, read, and decode the information on a medium, they must know the file allocation method to properly reassemble the files. Descriptions on items such as volume and file structures are identified in media standards, such as ISO9660 for the CD-ROM. The operating system also controls a media's file allocation method and therefore, it is necessary to access operating system specifications to gather data on file allocation methods. Each operating system and medium combination uses a specific allocation method. Examples of some popular allocation schemes are the contiguous, linked, and indexed allocation methods. The contiguous allocation method requires each file to occupy a set of contiguous addresses on a disk. With linked allocation each file is a linked list of sectors and the sectors may be scattered anywhere on the disk. With the indexed allocation method each file has its own index block which is an array of disk block addresses [Settani, 1995]. The allocation method may also provide other valuable information such as distinguishing between the locations of data bytes and error detection/correction bytes.

Collecting and maintaining metadata on these four entities (data storage geometry, storage methods, encoding schemes, and file allocation methods) provides the keys to recovering data once an access system is no longer available for that media type. As hardware and software systems become obsolete, metadata is used to develop hardware and software systems to recover data and migrate it to currently accessible storage media

## FILE FORMAT METADATA

The first step in gathering file format information is to identify all of the applications used to create the digital documents which may need to be reconstructed in the future. This step includes both COTS and non-COTS applications. Gathering and cataloging metadata to reconstruct digital documents created with COTS and non-COTS applications is a time intensive and difficult task. However, it is necessary because many organizations use these applications to create and store digital documents.

The second step is to identify and catalog the objects that are supported by these applications. An object in a digital document can be text, graphics, audio, video, and any number of other structures that have been included by the document's creator. It is necessary for an interpreter to have the ability to identify the objects embedded in a digital document before the interpretation process begins. If an object is not properly identified then the document is uninterpretable.

Once the objects are identified, interpretation routines are created to present these objects in their original form on the current information system. Since objects are used over and over again by different applications, it is only necessary to create a routine to interpret and display that object once. A routine can be used to display an object regardless of the application used to create the digital document. For example, most digital documents support the use of text objects. Since text is used in multiple applications, it is only necessary to create a routine to handle a text object once. That routine can then be used to interpret and display text on the current system regardless of the software and hardware systems that were used to create the original document.

The final step is to identify and catalog the formatting structures implemented within each application. These formatting structures describe how objects are identified, formatted, and arranged within a digital document. In addition, this information describes how to determine page layout information including page size, margins, line spacing, tabs, fonts, and footnotes. This information must be maintained in a standardized form so that an interpreter can easily access it and switch between digital documents that were created by different applications. The formatting process may be made more difficult because there is no standardized way in which applications store formatting information. Applications disperse formatting information

1. throughout the document,
2. in designated locations within the document, or
3. in combinations of 1 and 2.

In addition, some applications store document files in an ASCII format while others opt for a binary format. Defining a standardized method to describe these currently non-standardized procedures is one of the goals of the DRS metaknowledge archive.

## VIII. DATA RECOVERY

Once it is no longer economically feasible to maintain antiquated hardware systems, it is necessary to implement an alternate method to maintain the ability to recover data from superseded media. If data are stored on an obsolete medium that is not accessible by current systems then the data must be migrated to a currently accessible media before document reconstruction can begin. That is, the data must be recovered.

Data recovery involves the use of the storage technique information gathered during the knowledge preservation process to recover data from an obsolete media. This information is used to modify or construct the equipment needed to migrate digital data from an obsolete medium to one that is currently accessible.

An example, of this usage can be depicted by data stored on punched cards. Punched cards pass through a punched card reader at the rate of approximately 1,000 cards per minute. As the cards pass between a light source and a row of photo-electric cells the location of the holes are detected and the pattern is transformed into electric signals which are sent to the computer and translated into machine language [Downing, 1986]. Because of advances in storage technologies, punched cards are seldom used as a storage medium today because they are slow, bulky, and cumbersome compared to modern storage media. Few organizations maintain punched card today. So, if stacks of punched cards were to be found and there were no punched card readers available to read the data, how could the data be read? First, the punched card storage technique information that was gathered during the knowledge preservation process is retrieved. Once the information is analyzed and engineers understand the way information is stored on a punched card, they may find that it is a simple task to reprogram a modern scanning device, such as those used in supermarkets or on assembly lines, to read the patterns of holes on a punched card. Therefore, a device can be modified to read, translate, and migrate the data on punched cards to a modern storage device without the need for an original punched card reader. There is no need to engineer a device to write punch cards because there is no desire to change the data. The need is only to read the data and migrate it to a currently accessible storage medium.

While this is a relatively simple example of how the storage technique information can be used, it demonstrates how easily yesterday's digital technologies can be more easily reproduced using today's digital technologies. Likewise, this same method could be used to manufacture readers for paper tapes, CD-ROMs, and other storage devices. If someone finds a CD-ROM disk in the year 2222, perhaps he or she will be able to take it to a DRS processing center to recover the data. Instead of building a CD-ROM drive, the processing center may simply use a high-tech "scanner" to scan the disk and identify the patterns of lands and pits burned into the disk's surface. Using the data gathered about CD-ROM storage techniques during the knowledge preservation process, an information system analyzes the location and patterns of lands and pits, identifies the file allocation system, processes the data, and then writes the files to a twenty-third century storage device.

## IX. DOCUMENT RECONSTRUCTION

If digital documents are stored in superseded formats, then they must be reconstructed. Document interpreters, which accomplish reconstruction, are either

1. trained technicians or
2. software applications that use file-formatting information to reconstruct digital documents.

The DRS relies upon the file format descriptions gathered during the knowledge preservation stage to describe how the original software interpreted files. These file format descriptions identify the information, such as character sequences (and their locations if they are position sensitive), that identify data objects and specify formatting operations within a digital document.

Table 1 contains examples of the character sequences used by three different applications to perform **bolding** operations on text.

Table 1. Example Character Sequences for Bold

| **Software Application** | **Begin Bold** | **End Bold** |
|---|---|---|
| WordStar® | 02 | 02 |
| Ami Pro® | 3C 2B 21 3E | 3C 2D 21 3E |
| HyperText Markup Language | 3C 42 3E | EC 2F 42 3E |

For example, when an interpreter is reconstructing an Ami Pro® 3.1 document, the character sequence (hexadecimal values) “3C 2B 21 3E” specifies to the interpreter that all characters following this sequence need to be bolded. Likewise, the character sequence (hexadecimal values) “3C 2D 21 3E” signals the interpreter to stop the bolding process.

This view of how file format information can be used is simplified, but it demonstrates the types of information that need to be collected and stored to aid document interpreters in the reconstruction of all types of digital documents. In addition to identifying text-based objects and operations, character sequences are used to identify other objects imbedded within digital documents.

# X.THE DIGITAL ROSETTA STONE MODEL

The complete DRS model is shown in Figure 1. The DRS model can be represented in three stages:

1. knowledge preservation,
2. data recovery, and
3. file reconstruction.

The first stage of the model represents the knowledge preservation process. Preservation is the foundation upon which the DRS depends. During preservation the data needed to support data recovery and document reconstruction is gathered and stored in the metaknowledge archive.

The second stage of the model is the data recovery process. Data recovery uses the knowledge of storage techniques to extract a digital document's bit stream from an obsolete storage device and then migrates the bit stream to a currently accessible storage device. Once a digital document's bit stream is recovered, the bit stream is advanced to third stage.

The third stage of the model is the file reconstruction process. Document reconstruction uses the knowledge of file formats to interpret the bit stream and display the document in its original form. Upon completion of the reconstruction process, the final product is a reconstructed digital document that appears in its original form.

## EXAMPLE

The theory behind the Digital Rosetta Stone (DRS) can be demonstrated using an 8-track punched paper tape (8-TPPT). The 8-TPPT technology was widely used during the 1960s and 1970s. This technology was developed before industry standards were the norm and therefore, this technology is largely proprietary. Finding information on the 8-TPPT coding scheme was very difficult. While doing research for this paper, we contacted the technical support and archive sections of the IBM Corporation to get some information on 8-TPPT equipment. Unfortunately, we were told that IBM no longer supported this technology and does not maintain any information in its archives about it. However, some functional 8-TPPT readers still exist.

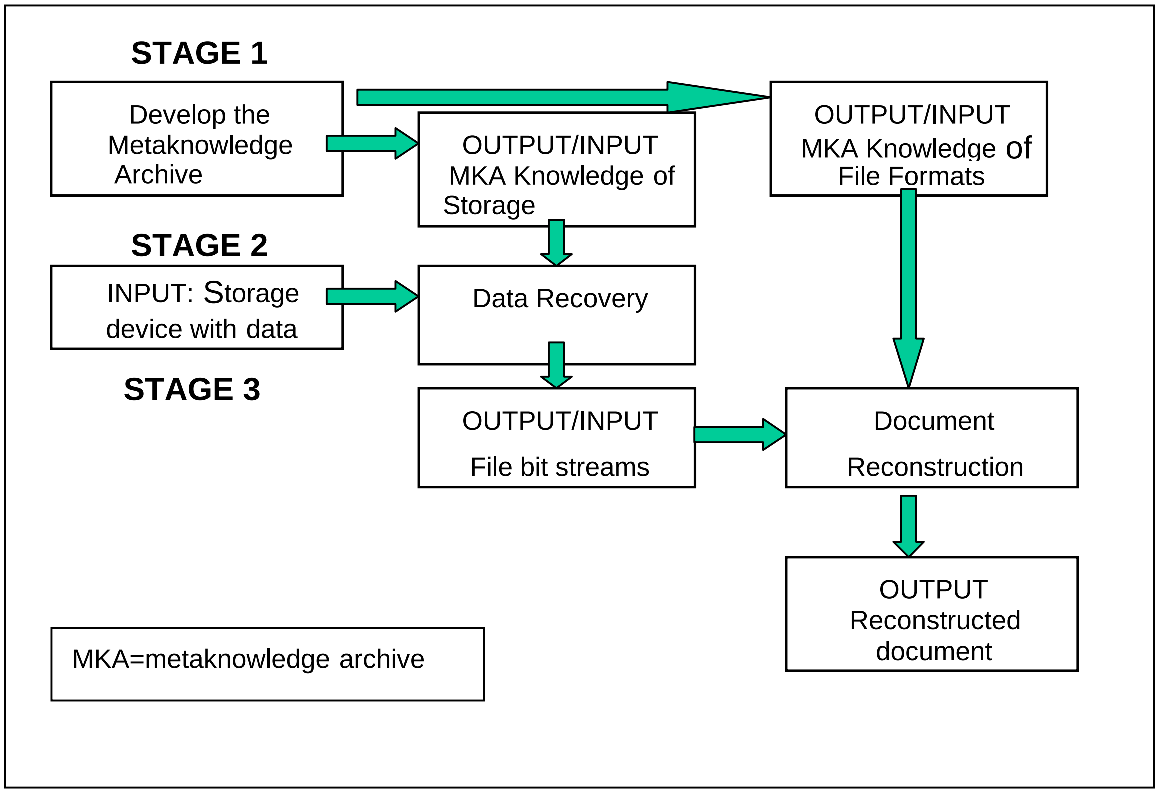
STAGE 1
Develop the Metaknowledge Archive
OUTPUT/INPUT MKA Knowledge of Storage
OUTPUT/INPUT MKA Knowledge of File Formats
STAGE 2
INPUT: Storage device with data
Data Recovery
STAGE 3
OUTPUT/INPUT File bit streams
Document Reconstruction
OUTPUT Reconstructed document
MKA=metaknowledge archive

Figure 1. Digital Rosetta Stone Model

After being unable to locate a listing of the character-coding scheme, several aging data processing books [Awad, 1971, Langenbach 1968, Nashelsky 1972, and Williams 1965] were consulted to find the information. All of the information concerning the 8-TPPT used in this example was compiled from these sources. While much of the coding scheme was obtained from these books, the set is far from complete.

The 8-TPPT stores data sequentially along the length of the tape. Individual characters are stored vertically on the tape in eight channels. The eight channels represent seven data channels and one check (or parity) channel. From the least significant bit to the most significant bit these channels are identified as 1, 2, 4, 8, Check, “O”, “X”, and the End of Line (EL). An example of

8-TPPT can be seen in Figure 2. Notice that unlike today, the check bit is not the most significant bit, but instead is in the fifth bit position.

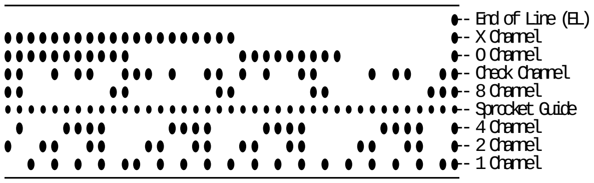

Figure 2. Example of 8-Track Paper Tape

Data are stored in the eight channels as follows:

- A punch or combination of punches in channels 1, 2, 4, and 8 represent numeric characters
- A punch in the Check channel is only to be used as a parity check (odd parity is generally used)
- A punch in the “O” and “X” channels are used in combination with channels 1, 2, 4, and 8 to define alphabetic characters, symbols, and other functions such as shift to upper case, shift to lower case, or stop
- A punch in the EL channel represents the end of a line and performs the same function as the return key on a typewriter

The patterns for upper case and lower case alphabetical characters are identical because the equipment used to print documents stored on 8-TPPT operated in a fashion similar to typewriters. That is, shift keys were used to define the difference between upper and lower case characters. Once a shift to upper case symbol was encountered, the type basket was shifted to the upper case position. All of the characters that followed were typed in the upper case mode, until a symbol was encountered to shift back to lower case. The shift from upper case to lower case mode, and vice versa, provided the ability to use an identical bit pattern for two separate symbols. Knowledge about shifting together with other meta-information about 8-TPPT is necessary to recover information from 8-track paper tape. [Nashelsky, 1972].

Upon examining the media storage technique information on 8-TPPT in the DRS, engineers find that they can reprogram a modern day scanner to a modern storage device. As the 8-TPPT is scanned, it logically partitions the horizontal tracks and vertical byte regions of the tape. An algorithm analyzes the data regions of the tape and converts the regions with no holes to a 0 and converts the regions containing holes to a 1. The bytes are then assembled into a bit stream, and migrated to a currently accessible storage medium. Once the bit stream is transferred to an accessible medium, it can be interpreted using 8-TPPT file formatting data that has been preserved in the metaknowledge archive. Using information from the metaknowledge archive, an interpretation algorithm reads the bit stream from the advanced medium and breaks the bit stream into 8-bit bytes.

The algorithm performs an error checking routine based on the fifth bit of the 8-bit byte to ensure that the integrity of the data was not compromised. Once error checking is complete, the 7-bit characters are mapped to the 8-bit character codes that can be displayed by the current system. When mapping the 8-TPPT's 7-bit characters to the character codes used by the current system it is necessary to use a translation table which maintains two translation schemes--one for upper cased characters and one for lower cased characters. That is, the same code used for the character "A" (0110001) was also used for the character "a" (0110001). The difference in character case was determined by the position of the type basket. Therefore, the algorithm translating the character set will have to track the position of the type basket and translate the characters appropriately.

Once the character set is translated the document can be printed. However, this is not as easy as it sounds. Many modern word processing operations, such as bolding, centering, and underlining are transparent to the document creator. However, the keyboarding techniques of the 1960s and 1970s were not as convenient. For example:

- To bold text, an individual had to type the text to be bolded, backspace to the beginning of that text, and then retype over the text.

- To center text an individual had to tab to the center of the page, backspace one-half of the total number of characters to be centered, and then type the text.
- To underline text an individual had to type the text to be underlined, backspace to the beginning of that text, and then use the underscore key to underline the text.

Therefore, to reconstruct these documents accurately, algorithms have to identify and translate these types of operations.

After all of this knowledge is brought to bear, a document stored 40 years ago can be recovered and printed. In the future, we will have many more difficult tasks of digital document reconstruction. The DRS can be a significant agent in helping to ensure that we do not lose our ability to read our own history.

## XI.RECOMMENDATIONS FOR FUTURE WORK

We have explored the looming problem of the loss of long-term access to many of our stored digital documents, and presented a model for capturing and maintaining the knowledge necessary to read and reconstruct those documents. However, we recognize that our model is preliminary, and that there is much work to be done if this model is to become a reality. First, we believe that for this model to be useful, it must be further developed, particularly with regard to the specific types of meta-knowledge that will be necessary to make it a success. Then, it would be valuable to develop a prototype of this concept to further test the value of the model for wider application. Finally, we recognize that this model is most likely to prove useful for those static types of digital documents that are basically cognates of paper-based documents. More complex digital documents will likely be very difficult to capture in this model. An important area for future research will be to try to identify the boundaries for the successful application of this model.

## XII CONCLUSIONS

In this paper we examined the problem of maintaining long-term access to digital documents. We reviewed the methods suggested by others, and combined them with additional ideas to create a model we call the Digital Rosetta Stone (Section X). The Digital Rosetta Stone describes a method by which we will be able to maintain long-term access to our increasing repositories of digital documents.

The development of a DRS will be a time intensive and expensive task. Consider the vast number of research projects, books, and museums that maintain access to our written history. The mechanics of the written language change slowly over decades and centuries. However, new technologies for capturing and storing digital documents are evolving faster than ever. This rapid development calls for the preservation of the vast amounts of digital knowledge that has been and is being created. However, unlike written documents, the preservation of digital documents also requires the preservation of the knowledge and technology necessary to access these documents. The Digital Rosetta Stone presents a model for achieving that end.

Editor's Note: This paper was received on May 19, 1999. It was with the author four months for one revision. It was published on January 31, 2000.

## REFERENCES

Awad, Elias M (1971) *Business Data Processing, Third Edition*. Engelwood Cliffs, N.J: Prentice-Hall, Inc., 1971

Dollar, Charles M. (1992) *Archival Theory and Information Technologies: The Impact of Information Technologies on Archival Principles and Methods*. Publications of the University of Macerata, Macerata, Italy, March

Downing, Douglas and Michael Covington (1986) Dictionary *of Computer Terms*. Hauppauge, New York, Barron's.

Langenbach, Robert G. *Introduction to Automated Data Processing*. Engelwood Cliffs, N.J.: Prentice-Hall, Inc., 1968.

Martin, James. *Information Engineering Book I Introduction*. Englewood Cliffs, New Jersey: Prentice Hall, 1989.

Nashelsky, Louis (1972) Introduction *to Digital Computer Technology, Second Edition*. New York: John Wiley and Sons.

National Research Council (1995) *Study on the Long-term Retention of Selected Scientific and Technical Records of the Federal Government Working Papers*. Washington, DC: National Academy Press.

Norton, Peter, Lewis C. Eggebrecht, and Scott H. A. Clark (1995) Peter *Norton's Inside the PC, Sixth Edition*. Indianapolis, Indiana: SAMS Publishing, 1995.

OASD (Office of the Assistant Secretary of Defense) (1995) *Automated Document Conversion Master Plan, Version 1*. Washington, DC: Department of Defense

Peterson, Del (1991) "Case Study: Improving Customer Service Through New Technology," *Journal of Information Systems Management*, (8)2 pp. 28-35 Spring 1991.

Rothenberg, Jeff (1995) "Ensuring the Longevity of Digital Documents", *Scientific American*, (272)1 pp. 42-47, January

Settani, Joseph A. (1995) "Making the Jump from Paper to Image," *Managing Office Technology,* (40)4, pp. 15-28, April .

Williams, William F. (1965) *Principles of Automated Information Retrieval*. Ontario, California: The Business Press

## ABOUT THE AUTHORS

Alan R. Heminger has been an Associate Professor of Information Resource Management at the Air Force Institute of Technology (AFIT) since 1994 and the program manager of the AFIT Graduate Information Resource Management Program since 1997. From 1989 to 1994, Dr. Heminger was an Assistant Professor of Management Information Systems at the Indiana

University School of Business. He received a Ph.D. in Management Information Systems from the University of Arizona in 1988.

Captain Steven B. Robertson earned a Bachelor of Science degree in Computer Science from the University of South Carolina in 1990 and a Master of Science degree in Information Resource Management from the Air Force Institute of Technology in 1996. He is currently the Mission Systems Flight Commander for the 325th Communications Squadron at Tyndall Air Force Base, Florida.

Communications of the Association for Information Systems