Study of the Use of the Presidential Electronic Records PilOt System (PERPOS): Final Report

William Underwood
Sandra Laib
Marlit Hayslett-Keck
Matthew Underwood
David Roberts

PERPOS Technical Report ITTL/CISTD 02-4
December, 2002

Georgia Tech Research Institute
Georgia Institute of Technology
Atlanta, Georgia

The Army Research Laboratory (ARL) and the National Archives and Records Administration (NARA) sponsored this research under ARL Contract No. DAKF11-97-D-0001, Task Order 64 (11 MAR 2002 - 31 DEC 2002). The findings in this paper should not be construed as an official ARL or NARA position unless so indicated by other authorized documentation.

Abstract

The objective of the research reported here was assessment of the information technologies used in the archival processing tools developed during Phase I research through use and evaluation of the tools by archivists. A number of refinements have been made to the archival tools. The separate tools for file system filtering, arrangement, preservation, review and description have been integrated into a single application—the Archival Processing Tool (APT). The capability to identify file types has been extended from 35 to over 170 file types. The filter used to separate operating system and application software from user-created files was changed from filtering on file names and lengths to filtering on the SHA-1 hash codes of files. File systems are packaged in TAR files with metadata describing the files and the provenance and scope and content of the file system (record series). The integrity of the files in a TAR file is ensured through use of Secure-Hash Algorithm-1 hash codes.

An experiment was conducted to evaluate the effectiveness of three document retrieval technologies—Boolean, statistical, natural language and knowledge based—in finding electronic records relevant to a FOIA request. The natural language and knowledge-based retrieval technology based on lexical subsumption and relaxation labeling, while not performing as well overall as the statistical technique, out performed the statistical technique when the request was for specific information, and the query involved just a few words.

Additional requirements for processing of records created on legacy hardware and operating system platforms with legacy application software have been identified. A system to support archival preservation must support conversion of obsolete file formats, which do not have viewers, to current or standard formats; recovery of corrupt files; extraction of files from archive and self-extracting archives; and decryption of password-encrypted files. Self-extracting archive files must be distinguished from other executable files, because they may contain user-created records. Furthermore, they should not be directly executed to extract the files, as they may overwrite files of the same name in the directory containing the self-extracting archive file, and the extracted files may also contain viruses. Accessioned files should be automatically checked for security markings to ensure that security classified records do not occur in accessioned record series presumed to be unclassified.

A pilot study was designed and the archival processing tools deployed at the Bush Presidential Library and in a computer laboratory at Archives II. However, archivists have not exercised all of the functions of the archival processing tools so the pilot study needs to continue.

The Archival Processing Tool supports many of the tasks involved in processing records created on legacy personal computers. However, to further improve processing performance, task-oriented analysis and design should be investigated as a methodology for identifying the data, information, knowledge and methods needed to support archivists in accomplishing archival tasks such as FOIA and PRA review and description.

Table of Contents

Introduction

The overall objective of the PERPOS project is identify, apply and assess advanced information technologies that can support archivists in processing electronic records created on personal computers. During Phase I of this research, software tools and technologies were developed that support the accession, arrangement, review and description of records stored in DOS and Windows file systems. During this phase of the research, archivists began to use these tools to process personal computer records from the administration of President George H. W. Bush. Feedback as to the tools utility in supporting actual processing of electronic records will enable better definition of functional and other requirements for processing electronic records and determination of whether the advanced technologies actually improve the archival process.

The purpose of this report is to summarize the results of research in the use of the tools by archivists and the refinement of these tools to better support archival decisions and processing of electronic records. In the next section, refinements made to the archival processing tools are described. This includes extensions of the file type identifier, the file system filter, the archival information packages, and support for digital preservation. Section 3 summarizes the results of an experiment in which three document retrieval technologies were evaluated as to their average precision in finding electronic documents relevant to FOIA requests. The fourth section describes the research design of the pilot study, a secure web portal for sharing tools and data among researchers and archivists, and the results of the initial use of the tools by archivists. The final section summarizes the results of the current research and discusses additional research issues.

Archival Processing Tool

The separate tools developed during Phase I research for file system filtering, arrangement, preservation, review and description were integrated into a single application interface—the Archival Processing Tool (APT). An on-line Help and User's Guide were also constructed for the APT [Underwood et al, 2002a].

The user interface to the APT is shown in Figure 1. A test suite of PC file types has been loaded. The *Activity* pull-down menu shows activities (or modes of operation) that the tool supports. In the *Explore* mode, the archivist can perform any of the operations common to all the activities. This includes opening a file system, opening and viewing a file, viewing a file's properties, closing or saving a file system, associating a viewer with a file type, and choosing another activity. The *Filtering* activity supports separating the files of a DOS or Windows file system into user-created files and operating system and software application program files. The *Arrangement* activity supports reordering files within directories, moving a file to a different directory, creating directories, and renaming directories.

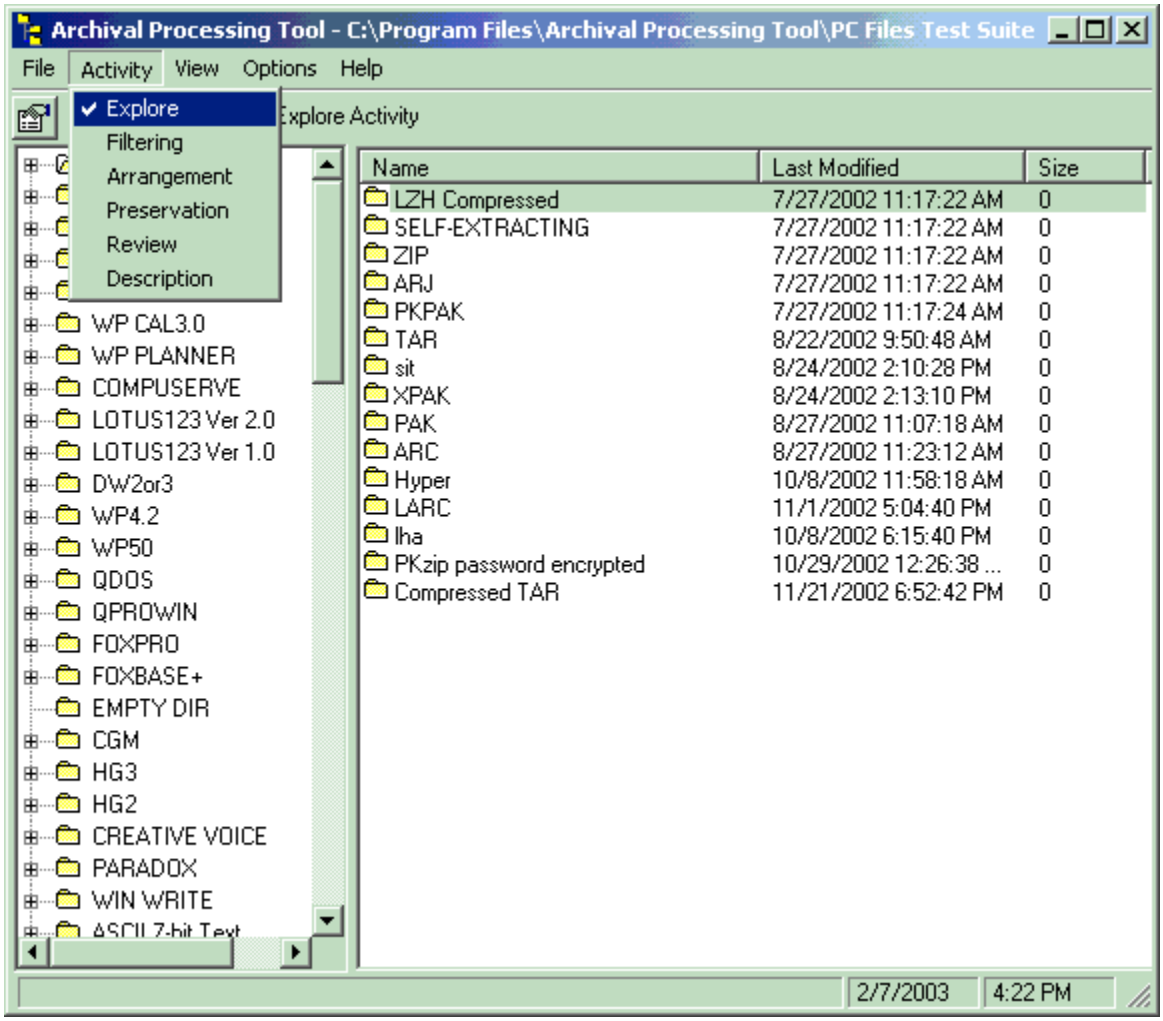

Figure 1. User Interface to the Archival Processing Tool.

The *Preservation* activity supports transformations of the original file so that the file can be viewed. These transformations include extracting files from archive files, recovering passwords from password encrypted files and decrypting those files, recovering corrupt files, and transformation of files with obsolete formats to current or standard formats.

The *Review* activity supports review of records for Freedom of Information Act (FOIA), Presidential Record Act (PRA) and Donor access restrictions. It supports opening, withdrawing or redacting the reviewed records. It also supports transfer out of the system of non-records such as Personal Misfiled records.

The *Description* activity supports archival description of the file system (record series), namely its provenance (organization, office, person/title) and business context (scope and content note).

File Type Identifier

To know which program should be used to view a file, it is necessary to know the file's type. File extensions alone are not sufficient to identify a file's type. For instance, many word-processing applications use the file extension *doc*, so the file extension is ambiguous with regard to file type. Furthermore, document composers often used the DOS filename plus extension to mnemonically identify a file, and sometimes the extension could be identical to that of another file type.

During Phase I research, a file type identifier was developed that recognized 35 file types. [Underwood et al, 2001]. The file identification (or classification) technique used is similar to that of the UNIX *file* command, which identifies files based on the magic number or file pattern of file formats. Many binary files have a so-called magic number (or file signature) at the beginning of the file to indicate its type to an operating system or application program. Some magic numbers are actually strings, for instance, the '%!' at the beginning of PostScript files. For those file types that do not have a unique magic number, or no magic number at all, there is usually a pattern of data types in the file format that enable the unique identification of the file type.

While the APT file identification technique is similar to that of the UNIX file command, the APT file identifier is more accurate than the UNIX file command. This is largely due to attention paid to including enough features in the schema for distinguishing files to correctly disambiguate them, and including syntax as well as keywords in the identification of different kinds of text file types (e.g., DOS batch files, dBase IV program files).

There were user-created files in the file systems of the Bush hard drives that were not recognized by the file type identifier. There were also OS and application software files from the Bush PC file systems whose file type was not automatically identified. If they could be identified it would reduce the effort needed to the filter the OS and application files from the user-created files.

There may be self-extracting archive files in the file systems from the Bush hard drives. A self-extracting archive file is an executable program file that includes both an archive file and a program to extract the contents of the archive file. Self-extracting archive files could contain user-created files. The file type identifier developed during Phase I research identified this type of file as an executable file. It is necessary to recognize them as self-extracting archives of various types.

The number of file types recognized by the file type identifier in the APT (version 2.05) has been increased to over 170 files types [Underwood and Laib, 2002]. This includes more than 80 legacy file types that are typically user-created files (documents, databases, spreadsheets, calendars and graphics). It also includes 27 different archive file formats, of which there are 18 self-extracting file formats.

The file type identifier still does not identify all of the types of files that occur in the file systems from the Bush hard drives. An experiment in which the file type identifier was applied to the file systems of 100 of the Bush hard drives showed that it identified the file type of 90% of the files. The error rate in identifying files was very low, in the order of 1 per 10,000 files processed. The average time to identify a file on an 800 MHz Intel processor was 9 milliseconds.

55 additional file types were identified in the group of files whose file type was not identified. These consisted of font resource files (e.g., HP LaserJet soft font), application resource files (DisplayWrite 4 printer function table), document files (e.g., Multimate Advantage 2), database files (Paradox 4 database and index), graphics files (e.g., Lotus 123 picture), and source program text files (e.g., Basic programs).

The file type identifier should be extended to handle these file types. The primary limitation on extensibility of the technique is identifying the formats of proprietary file formats for which documentation may not be readily available. For many of file types, it is necessary to empirically derive the magic number, file pattern, or file format from examples of the files.

To ensure that the file type identifier correctly identifies the file types that it purports to identify, it was necessary to create a test suite of examples of the types of files recognized by the file type identifier. The file type identifier is tested by loading the PC Files Test Suite into the APT, creating a filter that stops all file types that can be identified, and filtering the test suite on file type (see next section of this report). All file types in the PC Files Test Suite should be blocked.

A number of research issues arose in the process of creating the file type identifier [Underwood and Laib, 2002]. There is a need for a standard for naming file types. File extensions are inadequate. MIME content types and subtypes are adequate, but result in an overloading of the application content type, and extensive use of the octet-stream subtype.

The magic numbers or file characteristics necessary to identify a file should be represented in a standard format that is easily extensible. The representation of the UNIX *magic* file used by the *file* command is one possibility.

There is a need for a standard file format description language for representing the file formats of electronic records in order to support software migration or format conversion. The EAST (Enhanced ADA Subset) data description language is a possibility.[1] CNES is developing a tool called EXTRA that converts EAST descriptions to XML. To facilitate construction of viewers for the file format, or conversion from one file format to another, it is also necessary to be able to characterize the semantics of the file format.

[1] CCSDS (2000). *The Data Description Language EAST Specification (CCDDS 644.0-B-2)*. Blue Book. Issue 2. November. Also ISO 15889:2000 Space data and information transfer systems -- Data description language -- EAST specification.

File System Filter

During Phase I research, a file system filter was developed that used file types and file name and length to separate operating system (OS) and application software files from user-created files [Underwood et al, 2001]. While the filter has been effective in identifying unique files by file name and length, there are obviously cases where this criterion might fail.

File name alone does not uniquely identify a file. The same file might be given a different name at different times. Recall the number of files named readme.txt that have different contents. File name and file length do not uniquely identify a file. For instance, two text files of the same length, could have the same file name, but have different contents. File name, file length and file date do not uniquely identify a file. For instance, two copies of the same file might have the same file name and length, but be saved on different dates.

The best candidate for uniquely identifying a file is the bit string of the file itself. However, even this does not uniquely identify a file, because the same file might have a different semantics with respect to different programs that use it as data. The file bit string might have a different semantics under different coding systems, e.g., DOS Extended ASCII, EBCDIC or ISO 8859-1. Also, consider an XHTML document. If it is served as text/xhtml, the receipient is obligated to process it as XML, performing well-formedness and perhaps validity checking, namespace processing, and a small number of other tasks. If the same document is served as text/html, the interpretation of those bits is governed by the HTML RFC which has no provision for XML namespace processing or well-formedness.

It is the bit string of a file and contextual information, which can be represented as metadata and methods, that uniquely define a digital object and its behavior. A new version of the file system filter has been developed that uniquely identifies files that are OS or application software files in terms of the bit string of the file, the context of the file, the software application version that created it, and the operating system on which the software application executes [Underwood and Laib, 2002].

Bit string comparison of files from a file system against millions of files in a reference set (filter) is computationally inefficient. Comparison of the hash codes of a file against the hash codes of the files in a reference set is much more efficient. The hash code algorithm used is Secure Hash Agorithm-1 (SHA-1). The hash code is saved with metadata about the file in a filter database. Figure 2 shows a portion of the filter database.

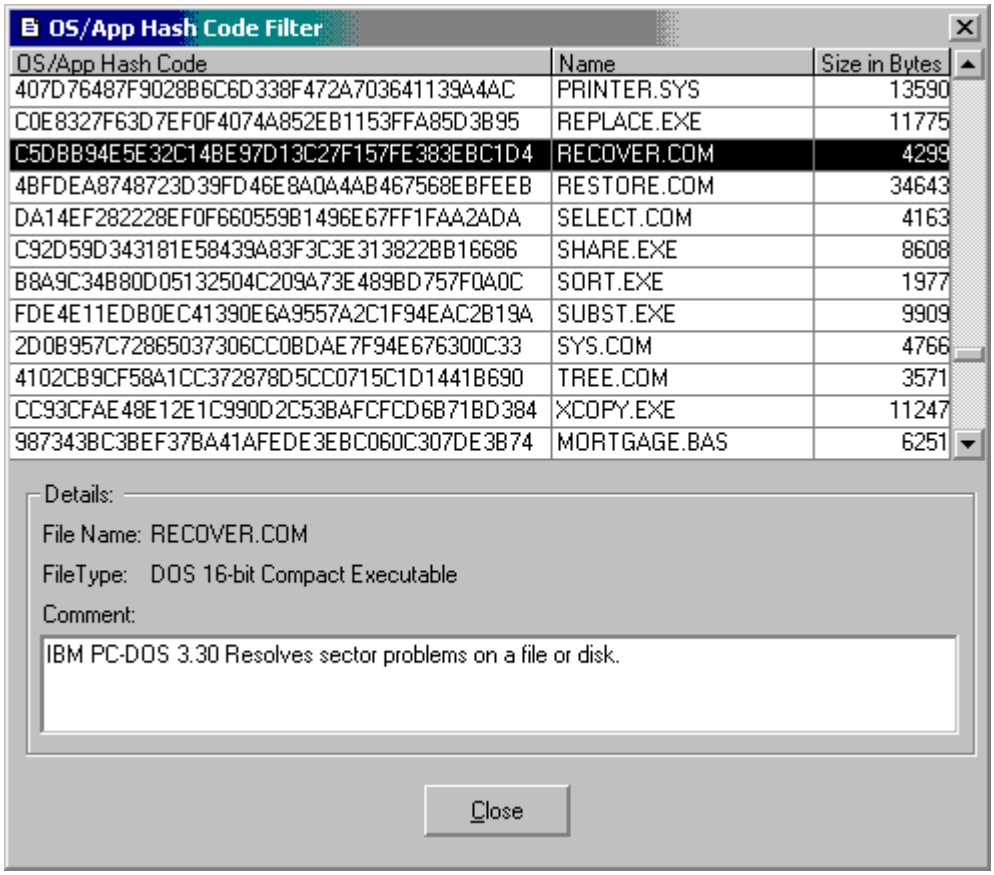

Figure 2. Display of the OS/App Hash Code Filter.

Since there is the possibility of collisions (different files have the same hash code), a second pass is made through the blocked files comparing the bit strings of those files that have the same hash code. This technique reduces to zero the risk of concluding that a file is a system or application file, when it was actually a user-created record.

Another reason for choosing this approach is that NIST has created the National Software Reference Library (NSRL) Reference Data Set (RDS) that includes the hash codes of operating system and application software files and metadata about these files.[2] The NSRL RDS includes CRC32, MD4, MD5 and SHA-1 hash codes for each operating system or application software file in the repository. Release 1.0 contained SHA-1 hash codes of 3,001,846 unique files from versions of approximately 1450 operating systems or products.[3]

[2] G. E. Fisher, Computer Forensics Guidance. Software Diagnostics and Conformance Testing Division Information Technology Laboratory, National Institute of Standards and Technology, Nov. 2001. www.nsrl.nist.gov/

[3] http://www.nsrl.nist.gov/

The NSRL Reference Data Set (version 1.0) was converted into a file filter and used with the filtering function of the APT to filter the contents of 20 of the Bush PC file systems.[4] Figure 3 shows the results.

Bush HD ID	NSRL RDS Filter		Custom Filter on File Types and Hash Code	
	Blocked Files	Passed Files	Blocked Files	Passed Files
1	7	3381	594	2794
11	7	2498	711	1794
14	7	399	387	19
15	42	3306	2683	665
22	14	2204	1185	1033
23	7	539	394	152
26	24	442	422	44
31	7	924	894	37
34	174	491	486	179
35	7	728	399	336
37	0	688	656	32
0039	11	936	795	152
70	7	364	367	4
71	9	367	376	0
0071	8	436	365	79
74	7	355	341	21
0105	60	714	422	352
0375	59	596	257	398
0536	7	258	214	51
0777	36	1365	754	647

Figure 3. Comparison of NSRL RDS Filter and Custom Filter Based on SHA-1 and File Type

The filter using the NSRL RDS did not perform as well as the manually created custom filter because the RDS contains primarily files produced after the Bush Administration (1989-1993). It includes hash codes for dBase III Plus files and several versions of MS-DOS that were in the file systems of this experiment. However, there were files from 25 legacy products that appeared in the file systems that were blocked by the custom filter but were not included in the NSRL RDS filter. For instance, these products included: IBM DOS 3.3 & 5.0, Windows 3.0, Harvard Graphics (versions 2.3 & 3.0), Lotus 1-2-3 (versions 1.0, 2.01, 2.2), Quattro Pro 1.0, and Word Perfect (versions 5.0 & 5.1), However, the NSRL continues to add legacy and current operating and application files to the RDS,[5] so over time it will become a valuable file system filtering resource.

Another reason for computing the hash codes of files is to be able to verify the integrity and authenticity of user-created files over time. This is discussed in the next section.

[4] A utility program has been written that converts the RDS into the format of a file system filter.
[5] Release 1.4 is now available.

Archival Information Package

The Consultative Committee for Space Data Systems (CCSDS) has developed a Reference Model for an Open Archival Information System (OAIS).[6] It includes a logical model for supporting preservation of information in an archival information package. An *archival information package* (AIP) is a file containing packaging information, a package description, content information, and preservation description information (PDI). The PDI contains reference, provenance, context and fixity information. File systems are packaged as a single file for efficiency in storage and file transmission. Metadata about the contents of a file system is included in the package because it is not needed until the file is processed.

The APT packages a file system (record series) into a TAR archive file with a manifest file containing metadata describing characteristics of the directories and individual files. The metadata includes file type, access restrictions (open, closed, redacted), and reason for closure. Figure 4 shows a portion of the manifest file for a file system.

```
Manifest-Version: 1.0

Organization: Georgia Tech Research Institute
Organizational-Unit: ITTL/CSITD
Name-of-record-creator: "Underwood, W., Principal Investigator"
Series-Title: Test Data for File Type Identifier of Archival Processing Tool
Accession-no: 2002-10

Name: ARCHIVE\
FileType: Directory

Name: ARCHIVE\LZH Compressed\
FileType: Directory

Name: ARCHIVE\LZH Compressed\wsd2snd.lzh
SHA-Digest: BA8AD40483EFC72187BE7B89789199165E10B229
FileType: LHA Archive

Name: ARCHIVE\SELF-EXTRACTING\
FileType: Directory

Name: ARCHIVE\SELF-EXTRACTING\ARJ SFX Executable\
FileType: Directory

Name: ARCHIVE\SELF-EXTRACTING\ARJ SFX Executable\arj241a.exe
SHA-Digest: 53BEF16AB9D11F491E33DBF531A412CAC4EF3403
FileType: ARJ 1.0 Self-extracting Archive

Name: ARCHIVE\SELF-EXTRACTING\LHA SFX Executable\
FileType: Directory

Name: ARCHIVE\SELF-EXTRACTING\LHA SFX Executable\lha213.exe
```

Figure 4. Manifest File of a File System.

[6] CCSDS 650.0-B-1: *Reference Model for an Open Archival Information System* (*OAIS*), Blue Book. Issue 1, January 2002. This Recommendation has been adopted as ISO 14721:2002

During current research, the capability was added to include the hash codes of files, to describe in the manifest the computer platform on which the records were originally created (e. g, computer, operating system), provenance (organization, office, person/title), record series description, and file order within directory (date last modified, alphabetic by filename).

An accession number is added to the manifest when the file system is accessioned. The provenance information (organization, organizational unit, individual name of record creator) can be added/edited during accession or the description activity. The file type and SHA digest (hash code) of each file is included in the manifest. The SHA-digest is used to check the integrity of the files in the package.

The CCSDS Panel 2, XML Packaging Workgroup is conducting research leading toward an ISO standard for XML packaging of space data and records. Future PERPOS research will be directed to representing the provenance and file structure information as XML schema.

The procedure for packaging file systems of electronic records was analyzed using a formal method to determine whether it preserved the authenticity of the records in the package over time [Underwood, 2002a, 2002c].

Digital Preservation

A primary objective of the PERPOS research is to support archivists in gaining archival control of records in the Bush PC file systems in their native formats, not in consideration of transformations to these files that might be needed to preserve the records for the long-term. However, there are some files in the Bush PC file systems that cannot be displayed without some transformation of the original file. These files include archive files, password encrypted files, damaged or corrupted files, and files for which there are no viewers. The APT was extended to support preservation transformations needed to reproduce (view) these kinds of files [Underwood et al 2002a; Underwood 2002b].

Extracting Archived Files

User-created files must be extracted from archive files, e.g., PKZIP, ARC and ARJ files, in order to be viewed. A directory with the same name as the archive file and on the same path as the archive file is created. The archive file is copied to that directory. The files in the archive are extracted into that directory. The files are checked for computer viruses. The file types of the files are identified. The archivist can view the files. For each file, the manifest can indicate the name of the archive it was extracted from.

As discussed earlier, a self-extracting archive file is an executable program file that includes both an archive file and a program to extract the contents of the archive file. The files in a self-extracting archive file can be extracted by simply executing the file. However, self-extracting archive files should not be executed in context in an archival

system. If the self-extracting archive file is executed in context, an extracted file may be saved over another file of the same name. Furthermore, the extracted files need to be checked for computer viruses. Consequently, in the APT, self-extracting archives are passed to a program that ignores the executable header and extracts the files into a separate area. Then the extracted files are checked for computer viruses,

Recovering Password Encrypted Files

Some of the PC application software of the late 80's and early 90's included the built-in capability to encrypt a file using a password. During experiments in processing the content of the Bush hard drives, password-encrypted files were discovered, for instance, password-encrypted Word Perfect, Quattro Pro, and PKzip files. Since the National Archives has the legal and physical custody of these files and is responsible for their preservation, archivists need the capability to recover the password in order to decrypt the files. The recovery of a password for legitimate and practical purposes should be distinguished from cracking of a system or file password for illegitimate purposes such as theft or vandalism. However, the techniques are the same.

There are commercial-off-the-shelf (COTS) products that recover the passwords of password-encrypted files. One of these products, Password Recovery Toolkit (PRTK) from Access Data, was acquired for experiments in recovering passwords.

When the file type identifier of the APT identifies a file as being password protected, an archivist can make a copy of the encrypted file and ask the PRTK to recover the password. If the PRTK has a procedure for recovering passwords for files of that type, it will attempt the recovery. In the cases of a weak encryption method, the password may be recovered in a few seconds to a few minutes. Stronger encryption methods may require hours to days of processing to recover a password. One can open the file using a copy of the original application used to encrypt the file, and use the password to decrypt the file. The file can be viewed with Quick View Plus and copied back to the file system with a different file name, but associated with the original password encrypted file.

The password can be recorded in the manifest, as can a record of the fact that the file was decrypted. The copy of the encrypted file in temporary storage, not the original in the file system, is erased.

Recovering Corrupted Files

During filtering experiments with the Bush hard drives, a number of WordPerfect 5.*x* files were encountered that could be read and recognized by Quick View Plus as WordPerfect 5.x files, but could not displayed properly. For example, Quick View Plus states that it cannot display the file or displays a blank screen, even though the text is was actually all there in the file. In other cases, a "loop" occurs in the document so that

moving the cursor down brings you to an earlier part of the document. The file header or function codes in the text of the file were corrupted.

The File Doctor for WordPerfect 5.x was obtained and used in experiments to recover some of the corrupt WordPerfect files.[7] Other user-created file types including database, archive, image and audio/video files may need repair. File recovery tools for some of these file types were acquired and integrated into the Archival Processing Tool. When a file is repaired, the original file, the repaired file, and metadata about the recovery are saved in the manifest section for the original file.

Converting Obsolete File Formats to Current or Standard Formats

The Quick View Plus set of software viewers, which recognize the file formats of over 200 file types, is used to view the contents of the more than 50 user-created file types occurring in the Bush PC file systems. However, Quick View Plus is not a complete solution to the viewing of the files from the Bush PC's. Quick View Plus does not have viewers for two types of database files from the Bush PC files (Advanced Revelation and Borland Reflex). Only the vector graphics of Harvard Graphics charts are shown, not the charts colored with a palette. It does not display the memo fields of dbase database files or use the database index to display the records in a database in the proper order. A similar situation occurs for Paradox database files. It does not display Word Perfect notebooks or calendars in their original form. Hence, it is necessary to find or write additional viewers, or to consider some preservation transformation on these file formats in order to view the records. For instance, Advanced Revelation databases are stored in two files with the same filename and two different file extensions, lk and ov. There are conversion packages that run under Advanced Revelation that will read the two database files and convert them to comma-separated values, dbase, or XML formats. These are formats for which there are viewers.

Finding Documents Relevant to FOIA Requests

Archivists need to respond to Freedom of Information Act (FOIA) requests for electronic records that have not been systematically processed. Four text-based retrieval technologies—Boolean, statistical, natural language, and knowledge-based— that can be used to find electronic records that are relevant to FOIA requests were evaluated [Underwood and Underwood, 2002]. Document retrieval experiments were conducted with document retrieval systems corresponding to each of the document retrieval technologies. Queries used in the experiments were derived from actual FOIA requests submitted to the Bush Presidential Library. The experiments were conducted using the Bush Public Papers as a sample collection.

WebGlimpse was used as an example of a Boolean text-based retrieval technology;[8] Oracle Text's word search as an example of statistical search technology; and Sun's Nova

[7] WPMD 3.0 - The File Doctor for WPerf5.x, Shareware From Software by Seidman.
[8] Glimpse Documentation. http://webglimpse.org/gdocs/glimpsehelp.html

Precision Content passage retrieval system as an example of natural language and knowledge-based search technology.[9] Precision content retrieval is based on the concepts of lexical subsumption and relaxation ranking. Lexical subsumption is the relationship of generality between concepts in which a term X subsumes a term Y if X is more general than Y, or equivalently, if Y is more specific than X. Relaxation ranking is a method of rating retrieved passages in a passage retrieval algorithm by penalty scores that measure how much the conditions of an exact match with a query must be "relaxed" in order to accept the match.

Average precision is a good measure of the utility of a document retrieval system. Average precision combines precision, relevance ranking and overall recall. Average precision is the sum of the precision at each relevant hit in the hit list divided by the total number of relevant documents in the collection. The average precision of Oracle word search was .7620; Nova .6165; and WebGlimpse .5436.

To facilitate computing average performance over a set of topics, each with a different number of relevant documents, individual topic precision values are interpolated to a set of standard recall levels (0 to 1 in increments of .1). The interpolated precision at standard recall level *i* is the maximum precision obtained for the topic for any actual recall level greater than or equal to *i*. These values are plotted in Recall-Precision graphs. Figure 5 compares the performance of the three technologies with regard to interpolated average recall-precision.

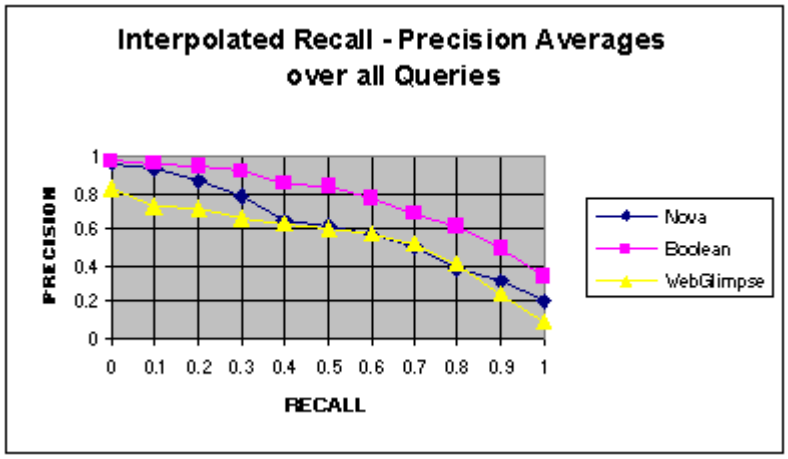

Figure 5. Comparison of Three Document Retrieval Technologies.

The results of the experiments were analyzed to explain the difference in performance for different topics. Oracle Text's word search had the best performance with regard to average precision, and especially for broad general queries with many alternatives. Nova's Precision Content Retrieval, while not performing as well overall, out performed Oracle Text on topics where the request was for specific information, and the query involved just a few words. WebGlimpse, using a Boolean search technology without ranking of hits, did not perform as well as the other search technologies.

[9] J. Ambroziak and W. A. Woods. Natural Language Technology and Precision Content Retrieval. Technical Report SMLI TR-98-69, Sun Microsystems Laboratory, December 1998.

The corpus size for this experiment was about 5200 files. To effectively evaluate text-based retrieval systems using ad hoc queries, one needs to use much larger corpora. For this and other reasons, the results of the experiment are not conclusive. For more conclusive results, the experiments should be conducted again using a larger corpus such as the Bush administration's PC file systems, which are an order of magnitude larger, or the Bush administration's e-mail records, which are three orders of magnitude larger.

Pilot Study

Archivists are using the Archival Processing Tools to process electronic records from the Bush administration. This will enable us to determine whether the tools actually support the day-to-day needs of archivists in processing electronic records, and to assess whether the technologies inserted into the activities actually improve the performance of the archival tasks.

Design of the Study

A study has been designed to answer questions such as the following [Underwood et al, 2002b]:

- How effective is the file system filtering tool in separating operating system and software application files from user-created files?
- How do the views of the files provided by the Quick View Plus viewers compare to the documents displayed or printed by the original software applications? Do they have the same content? Do they have the same page layout? Do they have the same font types and sizes?
- In what units should personal computer records be measured for each of the types of archival processing performed? Bytes (kilobytes, megabytes, gigabytes)? Files? Directories (folders)? TAR files? Pages?
- In what ways do the tools support or fail to support archivists in gaining intellectual and physical control of the files from the Bush hard drives?
- Do the tools support the normal work processes of the archivists at the Bush Presidential Library as they process the personal computer files?
- Are all of the operations (functions) provided in each tool needed?
- Are there any operations needed in the tools that are not provided?
- Is there any information that needs to be captured that is not provided for in the user interface?
- Is there any information that is being captured that is not needed?

Data will be collected during the pilot study that will enable the archivists and the computer scientists to answer these questions.

Each day, archivists at the Bush Presidential Library report in a Time and Production Record the volume of records processed by the type of processing performed, e.g., accession, arrangement, preservation, review, and description. Archivists using the tools will fill out an electronic version of this form specifically designed for the pilot study.

In processing the PC records, the archivists will identify additional archival and technological issues. For instance, what does the archivist do when there is a file read error for a user-created file? Is there another operation that the archival tools should provide? Is there a file type that is not automatically recognized by the tools? What are the archivists' and computer scientists' suggestions as to how to resolve these issues? The archivists and computer scientists will record their experience and questions that arise during the project in an electronic lab notebook (ELN). The electronic lab notebook provides a means for them to record notes of their observations as they go along, so that the accumulated experience can be reviewed and analyzed at the end of the study.

Secure Web Portal

A PERPOS Web Portal was constructed to enable archivists at the Bush Presidential Library and at Archives II to download the most recent version of archival tools, data and working papers and to communicate with researchers at Georgia Tech regarding any problems encountered during the pilot study. The archivists and researchers also have folders that can be used to document ongoing efforts and issues and to share files with others. The portal was constructed using Oracle Web DB services.

Access is controlled through a username and pass phrase. The username and pass phrase are encrypted during interaction with the web portal server. Fig. 6shows a part of the Web Portal interface.

The web portal has proven to be useful in providing updates to software versions, installation instructions, and data such as filters and test data. However, a higher degree of collaboration is still needed to address problems when they occur and to facilitate discovery of new issues. Collaborative tools need to be inserted into the work process in such a way that they can support collaboration among researchers and archivists who are not co-located (Roberts, 2002).

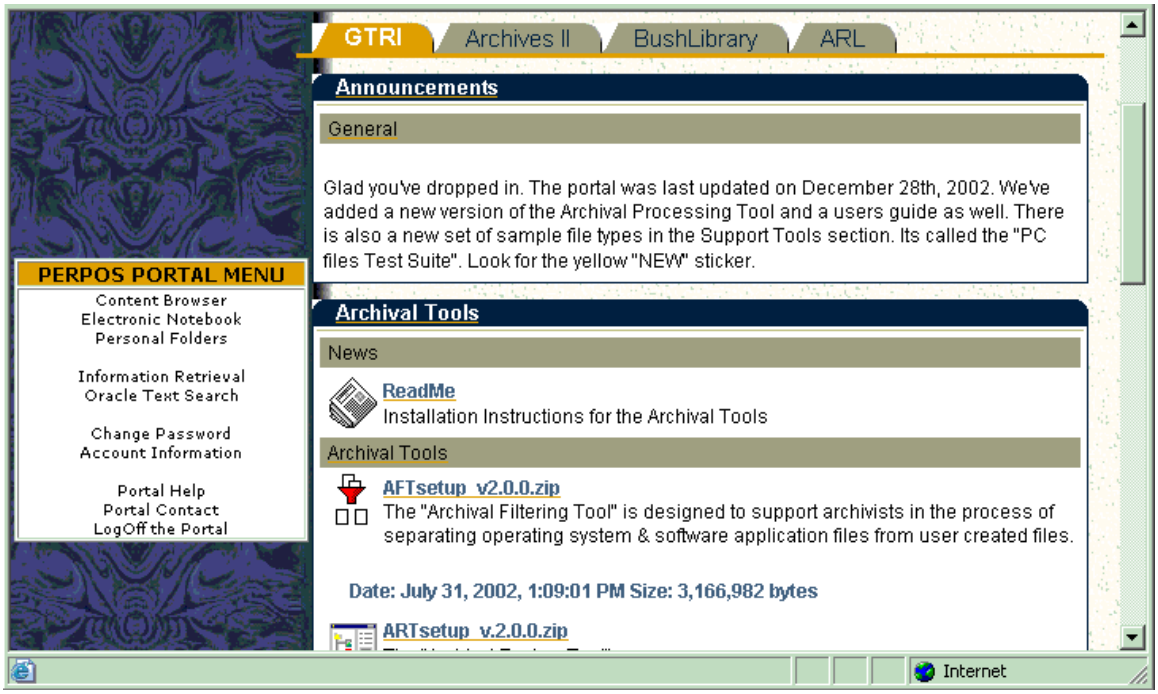

Figure6. PERPOS Web Portal Interface.

There has been one web server intrusion that was not detected and responded to in real-time. A firewall that only allows access to the portal by approved sites will address this issue in part. However, there is a need to investigate advanced information assurance technologies to mitigate threats to the portal (or to an archival repository) of physical and cyber attacks, human error, and hardware failure.

Initial Results of the Study

One of the first steps in the study was to load the entire set of Bush PC file systems onto a workstation at Archives II. During this process it was discovered that when the file systems had been copied from the Bush hard drives to a medium on which they could be preserved and processed, the directory dates in the file systems had being changed from the original date the directory was created to the date they were copied. This may be an artifact of the copying process that archivists want to avoid in the future. We did not want the directory dates to change again as they were copied onto the workstation. Since the file systems were in a FAT16 file system format, it was determined that they could be copied to a FAT32 file system format on the workstations using the DOS xcopy command without changing the directory dates.

The archival workstations at the Bush Presidential Library and Archives II are configured so that Norton Antivirus checks for computer viruses when any file is read or written. During the process of copying all of the file systems to the workstation, Norton Antivirus detected a computer virus in one of the files. It was quarantined. If the virus had been in a

user-created file, Norton Antivirus could probably have removed the virus. However, the virus was in an operating system file, so the file could be removed from the file system. The virus was a version of the cascade virus, not a dangerous virus, but one that should not be preserved in a collection of records, and not one that a Presidential Library would like to release into the wild at some time in the future.

During October, use of the Archival Processing Tool began at Archives II. An archivist used the tools to review some of the Bush PC file systems. A short time after the archivist began to use the APT, he discovered records in the file system of the hard drives that were not in compliance with national security requirements. This was an unexpected and unforeseen event. The Bush PC files are from offices where the PC's were used for unclassified work.

This event complicated the pilot study of the PERPOS archival tools. The data to be used for testing at the Bush Library and Archives II was no longer available. It was decided to test the presumption that the records in the Bush PC file systems did not include records with markings indicating a security-classified document. The Unix *grep* command searches one or more input files for lines containing a match to a specified pattern. *Grep foo file* returns all the lines that *contain* a string matching the expression *foo* in the file *file*.[10] Window's versions of *grep* will be used to search for files that match patterns corresponding to markings such as Confidential, Secret, and Top Secret, so that an archivist can review the documents to determine whether or not they are security classified documents.

This initial use of the APT points to the need to develop an information filter for electronic records that are presumed to be unclassified to check them for security-classification markings. This filter could be applied in the White House Office of Records Management before transferring the electronic records to the National Archives, or at the National Archives prior to loading them onto an uncertified workstation for processing, or both. Without such filtering of the files, and a hardware and software configuration that is acceptable to a certification authority, archivists may repeatedly find themselves in violation of security regulations.

Due to the security infraction, archivists at the Bush Presidential Library and Archives II have not yet had the time to fully exercise the operations of the Archival Processing Tools, e.g., to completely process a single file system. Consequently, the pilot use of these tools needs to continue in order to answer the posed questions.

[10] *Grep* is an unintuitive Unix command name. It seems to derive "from the qed/ed editor idiom g/re/p, where re stands for a regular expression, to Globally search for the Regular Expression and Print the lines containing matches to it." E. S. Raymond, *The New Hacker's Dictionary*, third edition, Cambridge: MIT Press, 1996, p. 226.

Conclusions and Research Issues

A number of refinements have been made to the archival tools. The separate tools for file system filtering, arrangement, preservation, review and description have been integrated into a single application—the Archival Processing Tool (APT). The capability to identify file types has been extended from 35 to over 170 file types. The filter used to separate operating system and application software from user-created files was changed from filtering on file names and lengths to filtering on the SHA-1 hash codes of files. File systems are packaged in TAR files with metadata describing the files and the provenance and scope and content of the file system (record series). The integrity of the files in a TAR file is ensured through use of Secure-Hash Algorithm-1 hash codes.

An experiment was conducted to evaluate the effectiveness of three document retrieval technologies—Boolean, statistical, natural language and knowledge based—in finding electronic records relevant to a FOIA request. The natural language and knowledge-based retrieval technology based on lexical subsumption and relaxation labeling, while not performing as well overall as the statistical technique, out performed the statistical technique when the request was for specific information, and the query involved just a few words. Another document retrieval experiment with a much larger collection is needed.

Additional requirements for processing of records created on legacy hardware and operating system platforms with legacy application software have been identified. A system to support archival preservation must support conversion of obsolete file formats, which do not have viewers, to current or standard formats; recovery of corrupt files; extraction of files from archive and self-extracting archives; and decryption of password-encrypted files. Self-extracting archive files must be distinguished from other executable files, because they may contain user-created records. Furthermore, they should not be directly executed to extract the files, as they may overwrite files of the same name in the directory containing the self-extracting archive file, and the extracted files may also contain viruses. Accessioned files should be automatically checked for security markings to ensure that security classified records do not occur in accessioned record series presumed to be unclassified.

The pilot use of the use of the tools at the Bush Presidential Library and Archives II has been initiated. Several additional archival requirements have been identified. Archivists should not presume that accessioned PC files are free of computer viruses. If they make this assumption, they run the risk of infecting the archives with potentially damaging computer viruses and the likelihood of releasing them back into the wild when the files are distributed. All workstations used for archival processing of electronic records must have computer virus detection software installed.

The Archival Processing Tool enables an archivist to open, withdraw, or redact electronic records, to extend or create titles for directories, and to summarize the contents of record series in scope and content notes. However, until an archivist finds the time to process the files, the repository has a low degree of intellectual control over the records. It is likely

that information extraction technology coupled with summarization technology could be used to identify document types (e.g., memo, correspondence, press release, agenda, speech), create preliminary folder titles for directories, and preliminary summaries for record series, thus giving an archive greater intellectual control over the electronic records. When an archivist gets around to systematically reviewing a records series, information could be extracted from the documents in folders and record series that would support the archivist's description task, and thus potentially reduce archival workload and increase throughput.

Currently, a Presidential Library's staff responds to a FOIA request by searching manually-created series title lists and folder title lists to identify records in unprocessed record series that might be relevant to the FOIA request. Text retrieval technologies were evaluated as to their precision and recall in finding electronic documents that might be relevant to a FOIA request. If information extraction technology could be used to automatically identify document type, addressee, author, date, subject or title and other attributes of an electronic record, a copy of the document could be marked up. Then text-based search to respond to FOIA requests could be augmented by structured text retrieval, which typically increases the precision and recall of text retrieval.

A method has been developed for filtering files in a file system into those representing electronic records and those that represent non-records. The filtering method is efficient and effective, but still relies heavily on the knowledge of the computer technician and/or archivist to initially recognize operating system and software application files and distinguish them from records. It is still dependent on the archivist's knowledge of document types, names and titles of personnel who might have created or received records, document content, and business context.

It may be possible to improve the file filtering technology by automatically creating the file filter. This would involve identifying document types and recognizing that a directory contained document types characteristic of user-created records rather than OS and application software. It would require the use of domain knowledge (e.g., person's names and titles). For instance, if a file is of document type memo and is to the President, Vice President or a member of the White House staff and is from the President, Vice president, or a member of the White House staff, it is a record.

Review of Presidential electronic records for FOIA and PRA access restrictions is an intellectually demanding task that requires page-by-page review of the records. Due to the increasing volume of electronic records from all branches of government, the need to review these records, and the cost of the limited human resources that can be applied to the review process, one should consider whether there are advanced technologies (natural language processing, case-based reasoning, and machine learning) that could increase throughput by supporting the review process. Task-analysis can be used to identify the kinds of knowledge and reasoning that archivists use to judge whether passages of documents or entire documents are subject to access restrictions. These findings might be used to develop a tool to support archivists in reviewing electronic records.

A distributed, heterogeneous computing environment for digital archives will be vulnerable to equipment failure, human error and physical or cyber attacks. Research is needed to demonstrate advanced information assurance technologies that are effective in mitigating these threats.

Results of experiments conducted in processing the Bush PC records point to the need for better description of records in file systems before they are transferred to the National Archives. There is an opportunity to reduce the archival processing workload by re-engineering the record life cycle with respect to records scheduling, disposition, transfer, and archival processing activities. In particular, tools are needed to support the packaging and description of electronic records transferred to the National Archives.

The Phase I PERPOS research was directed toward gaining archival control of records in their native file formats. The Quick View Plus set of software viewers, which display the contents of over 200 file formats, is used to view the contents of the more than 50 user-created file types occurring in the Bush PC file systems. The documents displayed with the Quick View Plus viewers should be compared to those displayed and printed with the original application to determine whether they adequately represent their original content, documentary form, and physical attributes.

There is a need for a standard for naming file types. File extensions are inadequate and while MIME content types and subtypes are adequate, they do not adequately characterize the wide variety of applications. There is also a need for a standard file format description language for representing the file formats of electronic records in order to support software migration or format conversion. A central registry/repository for file type names, magic numbers, and file format descriptions is needed.

Publications

Roberts, D. (2002) The Use of a Collaboratory for Electronic Records Archival Support, Research and Tool Development Coordination Draft, Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, August.

Underwood, W. E. (2002a) A step towards a logical theory of record integrity and authenticity. Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, March.

Underwood, W. E. (2002b) The Preservation of Legacy Personal Computer Records. Panel on Digital Preservation. *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital Libraries,* Portland, Oregon, July 14-18, pp. 365-367.

Underwood, W. E. (2002c) A formal method for analyzing the authenticity properties of procedures for preserving electronic records. *Proceedings of the 2002 International Conference on Digital Archive Technologies (ICDAT2002).* December 19-20, Academia Sinica, Taipei, Taiwan, pp. 53-64.

Underwood, W. E., Kindl, M. R., Underwood, M. G. and Laib, S. L. (2001) Presidential Electronic Records PilOt System (PERPOS): Phase I Report, Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, August.

Underwood, W. E., Hayslett-Keck, M. and Laib, S. (2002a) The Archival Processing Tool (APT): User's Guide Version 2.04. PERPOS Technical Report ITTL/CISTD 02-2, Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, November.

Underwood, W., Hayslett-Keck, M., Roberts, D., Underwood, M. and Laib, S. (2002b) Pilot Study of the Use of the PERPOS Tools for Archival Processing of Personal Computer Records, PERPOS Working Paper #8, Georgia Tech Research Institute, Atlanta, GA, June.

Underwood, W. and Laib, S. (2002) Identifying File Types and Document Types and Filtering Records from Legacy PC File Systems, PERPOS Technical Report ITTL/CISTD 02-1, Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, September.

Underwood, M. G. and Underwood, W. E. (2002) Evaluation of Document Retrieval Technologies to Support Access to Presidential Electronic Records, PERPOS Technical Report ITTL/CISTD 02-3, Georgia Tech Research Institute, Georgia Institute of Technology, Atlanta, December.