The International Journal of Digital Curation
Issue 2, Volume 4 | 2009

Creating Virtual CD-ROM Collections

Kam Woods and Geoffrey Brown,
Department of Computer Science, Indiana University

Abstract

Over the past 20 years, more than 100,000 CD-ROM titles have been published including thousands of collections of government documents and data. CD-ROMs present preservation challenges at the bit level and in ensuring usability of the preserved artifact. We present techniques we have developed to archive and support user access to a collection of approximately 2,900 CD-ROMs published under the Federal Depository Library Program (FDLP) by the United States Government Printing Office (GPO). The project provides web-based access to CD-ROM contents using both migration and emulation and supports remote execution of the raw CD-ROM images. Our project incorporates off-the-shelf, primarily open-source software. The raw data and (METS) metadata are made available through AFS, a standard distributed file system, to encourage sharing among libraries[1].

[1] This article is based on the paper given by the authors at iPRES 2008; received October 2009, published October 2009.
The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. ISSN: 1746-8256 The IJDC is published by UKOLN at the University of Bath and is a publication of the Digital Curation Centre.

Introduction

CD-ROMs present significant preservation challenges. At the bit level, the obvious technique is to create an (ISO) image of the standard ISO-9960 file system; however, CD-ROMs are subject to bit rot and generally do not provide checksum information to determine if an image is error-free. Building a viable archive requires the comparing of images from multiple instances of a single item. This can be viewed as an inverse of the problem solved by LOCKSS – lots of copies are required to create a single reference image which might itself be preserved through a system such as LOCKSS (Maniatis, Roussopoulos, Giuli, Rosenthal & Baker, 2005).

At the usability level, ISO images are large (up to Gigabytes), must be "mounted" to enable access to their contents, and often require software installation in order to use those contents. Furthermore, such required software is quickly becoming obsolete. We assume a use model in which most patrons will be satisfied with the ability to browse within a CD-ROM and easily access documentation and data in obsolete formats. For a minority, access requires mounting and executing software from the CD-ROM on a physical or virtual machine. Many items, although not the FDLP materials, require authentication in order to ensure that copyright restrictions are satisfied.

Our research has focused on handling of the FDLP document collection held by the Indiana University Libraries, one of 32 depositories in Indiana to which the Government Printing Office distributes select materials (Indiana State Library, 2006). This collection represents an ideal research workload because it is an important preservation target, it is temporally and technologically diverse, and it presents few copyright restrictions (Depository Library Council, 2006). The techniques we describe generalize to other CD-ROM based materials. Searching the Indiana University Library reveals more than 14,000 items. A similar search of the OCLC Worldcat system reveals more than 120,000 items.

The GPO has published approximately 5,000 unique CD-ROMs and DVDs created by various government agencies and has distributed these publications to various subsets of the 1,450 depository libraries. These collections contain fundamental information about the economy, environment, health, laws and regulations, and the physical and life sciences. The technological span of the collection ranges from items created for MS-DOS and Windows 3.1 – requiring execution of proprietary binaries – to recent items relying exclusively on commonly available commercial applications.

The FDLP is organized as a hierarchy of state-level regional repositories holding complete collections and all other repositories holding subsets[2] A patron wishing to access a particular item must first locate a repository holding the item and then obtain physical access to typically non-circulating materials. The libraries support physical access to CD-ROMs through "reference" workstations mandated by the GPO (MTR, 2005).

Our project is creating a virtual collection of CD-ROMs accessible from any Internet-enabled location. The collection is browsable via a web server and the CD-ROM images accessible through a distributed file system Andrew File System (AFS).

[2] About the FDLP http://www.fdlp.gov/home/about

In a typical use-case, a patron searches the database to find items of interest, browses those items to determine suitability, and mounts images on a physical or virtual workstation. It is anticipated that libraries will utilize standard virtual machine (VM) technologies to replace existing reference workstations. Our project includes script development to simplify the use of a VM to access the collection.

In contrast with traditional repository models, our objective is to enable libraries to integrate a collectively maintained "virtual collection" into existing collections. By retaining the images and metadata in AFS, libraries are empowered to pool otherwise disparate resources. The web-browsing capabilities can be seamlessly integrated into a library's infrastructure, while AFS provides the means to maintain and adjust access privileges over multiple Kerberos domains.

The remainder of this paper focuses upon the techniques and tools used to build a web-based project providing browsing and execution of CD-ROM collections. The technical discussion is divided into two sections. The first deals with accessing CD-ROM contents including file access, format identification, web-based browsing, migration, and the use of virtualization tools to support legacy executables within CD-ROM images. The second section includes CD-ROM image preservation and distributed image access. Throughout this paper we refer to ISO images which are the "bit-faithful" copies of CD-ROMs; ISO is short for ISO9660 which is the standard data format for CD-ROM contents (ECMA, 1987). We conclude with a discussion of related work.

File Access

Given a collection of ISO images of CD-ROMs, and the ability to read the files contained within these CD-ROMs, how can we ensure the continued utility of these files? As we discuss in the Image Access section, preserving raw access to these files is "easy". However, ensuring their continued utility in the face of obsolescence is hard.

The foundation for our experimental work is a basic web service that supports search, browsing, and migration to modern formats; however, this web service is intended purely as a demonstration vehicle. Our overall approach utilizes open-source software libraries and tools that can be integrated into existing collections. Indeed, the core web service is simple – requiring approximately 750 lines of Perl. However, in building this service we were forced to address fundamental issues involving file access (discussed further in the Image Access section), file format identification, browsing of files with hardwired context dependencies (e.g., HTML), and building reliable migration services. Because many of the FDLP items are dependent on proprietary or obsolete binaries, we assume that committed users will need to utilize emulation (virtualized) execution environments. A part of our research effort has explored the use of automation to simplify emulation-based access including the creation of automated installers for proprietary applications.

The remainder of this section is organized as follows. We begin with an overview of our web service to introduce the fundamental issues, including file browsing in the face of contextual dependencies. We then consider format identification, file migration, and finally emulation.

Web Service

The web service we developed to access the FDLP collections has a conventional user interface – a user finds items of interest through a search interface; these items are presented in a manner analogous to a conventional library catalog. The metadata listing for each item provides links supporting the browsing of ISO image contents or access to the raw image. Browsing within an image is analogous to a file browser with file title, type, size, and creation date. Individual files may be accessed in original format or migrated rendition.

The web service is driven from AFS-accessible ISO images and corresponding metadata in METS format[3]. The search indices and human-readable "catalog" pages are generated from the METS metadata through XSLT transformation. Browsing within ISO images is supported by separate binaries to identify formats, extract files and directories, and migrate files to modern renditions. This partitioning is intended to make integration of the underlying technologies into existing library collections "easy" – the web service is a relatively thin code veneer binding the raw data and metadata with tools supporting browsing.

An important design decision is that all ISO images and their constituent files appear to reside within a static file system hierarchy. In an earlier implementation, which was indexed by Google, we found it difficult to locate the context of an individual file. In our current implementation, the context of a file can be found by "walking" up the URL from file to enclosing directory to image, and ultimately to the catalog metadata describing an item. URLs for migrated renditions are encoded as HTML "gets" based on the original URL. For example, an Adobe PDF rendition of `../foo.doc` is accessed using `../foo.doc?migrate=pdf`. The listings for directories containing files which have migrated renditions provide appropriate icons for accessing those files. As illustrated above, the original rendition of a migrated file is easily located by dropping "?..." from the corresponding URL.

A significant problem with browsing arises from links. For example, an HTML file within some ISO image may refer to pictures or other HTML files. These links may be relative (e.g., foo/bar.html) or absolute (e.g., /foo/bar.html). Unfortunately, the latter implicitly refers to the root of the ISO image rather than the root of the web server. In our system we found it necessary to interpret and patch HTML files as they are served in order to ensure that the browsing experience works as expected. Unfortunately, such patches are not always feasible. For example, PDF files may have embedded links, or HTML files may use links within javascript. Hence, our patching is good, but imperfect.

We use Swish-e (Simple Web Indexing for Humans - Enhanced) to provide indexed search via the "title", "abstract", "subject", and "classification" categories drawn from the METS records associated with each ISO (Rabinowitz, 2004). A default query made in the web interface is searched by title only, although more sophisticated searches using Boolean operators, wildcards, or requesting specific SUDOC number(s) are also handled.

[3] Metadata Encoding and Transmission Standard (METS) http://www.loc.gov/standards/mets/

Each query returns a number of hits corresponding to catalog records. Selecting an individual hit returns a page providing the full formatted METS record, along with links to browse the contents of an ISO or download the full image. Information about current directory location and file format of any object within that directory is provided to Apache by a series of Perl CGI scripts processing the ISO using the `libiso9660` library and the Free desktop Shared MIME database (Leonard, 2008). Additional information about this method is provided in the Image Access section.

A user browsing the ISO is presented with a modified Apache-style listing for file objects within the directory hierarchy. For each object this includes an icon selected according to MIME type, a link to a migrated rendition (if available), name, modification date, and size. Formats for migrated renditions (including - primarily – HTML and PDF) are chosen for ease of access within a standard browser.

The web service described here is modular, designed to be used for standalone access to independent CD-ROM collections or integrated into existing archival systems. Migration services are therefore loosely coupled to the rest of the service, and may be run on a dedicated server. A simplified representation of the web service backend, along with a typical client setup as discussed in the Emulation section, is given in Figure 1.

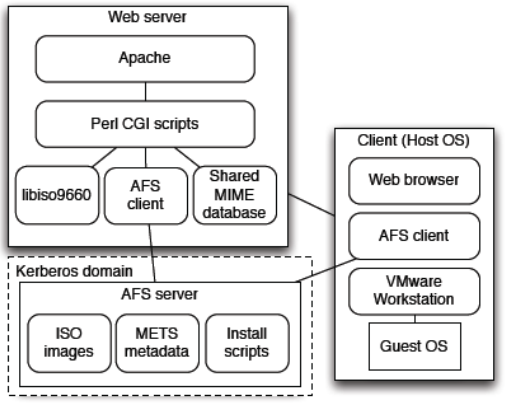

Figure 1. Web server overview.

Object Format Identification

Accurate file format identification is critical both to the presentation of sensibly marked document links by the web service and to the automated server-side migration services. Identifying specific files for which migrations can be performed is particularly difficult, since any file format identification scheme will generate a certain percentage of false positive hits for each document type. Our strategy uses a combination of existing open source tools for identification along with additional scripted tests and heuristics to provide breadth of coverage while tolerating failures gracefully.

We use the open source Shared MIME-info Database specification developed by the X Desktop Group for primary identification. In particular, we use the `libsharedmime` implementation found in the current distribution of the Gnome

desktop. This has a number of advantages over other available file format registries[4] [5]. It is production quality, fast, integrated into the Unix environment we use for migration services, has an easily customizable database, and produces succinct machine-readable descriptions of identifications. It remains in active development, and specialized (complementary) database updates for field-specific (for example, chemistry or GIS) file types are readily available.

File extensions and simple analysis for binary content are used as secondary identifying characteristics. This provides a degree of flexibility in handling the original file object. As an example, trials on the CD-ROM collection have indicated that both the Shared MIME-info Database and preservation-specific tools such as DROID will generate false positives or tentative hits for documents with the ".doc" extension containing some binary data that are not, in fact, Microsoft Word documents – or, in certain cases, cannot be migrated to a modern format using the OpenOffice document filters without damage or data loss. The secondary identifying characteristics allow for a more finely grained distinction between conversion failures and may generate fallback conversions when required (eg., text-only extraction for those office documents where binary content is mangled or cannot be appropriately identified). These generated materials are intended primarily to improve collection access rather than address the multitude of technical and long-term preservation issues presented by format migration (Council on Library and Information Resources [CLIR], 2000).

Migration

Our project tracks a diverse set of candidate file types for format migration. The web interface streamlines access by providing links to migrated renditions of original materials. Examples include Microsoft Office documents, Lotus 1-2-3 files, media items, and scientific binary formats. Migrated renditions are created on user request from the original document source to minimize error (Mellor et al., 2002). These migrated renditions are generated by a collection of open source tools if not previously cached. As previously discussed, contextual information is used to rewrite HTML sources where necessary for browsing, such as for archived websites with broken absolute site-internal links.

We use a collection of open source migration tools along with control scripts to create migrated renditions of documents in legacy formats. Our emphasis is on leveraging existing frameworks - the Shared MIME-info database, OpenOffice format filters, and the Python-UNO OpenOffice API bridge - and server-side scripting to provide both on-demand and batch migration paths for each selected format. Our approach tolerates and logs conversion failures in the background. The web service provides links to exactly those files for which successful conversions have been performed.

A Python framework coordinates both batch and on-demand conversion tasks. For batch conversions, a high-performance subprocess is called to rapidly generate a walk of the content within one or more ISO images. The results are filtered for the requested conversion formats, and written to a separate log for each ISO.

[4] The technical registry PRONOM http://www.nationalarchives.gov.uk/pronom/
[5] Global Digital Format Registry http://hul.harvard.edu/gdfr/

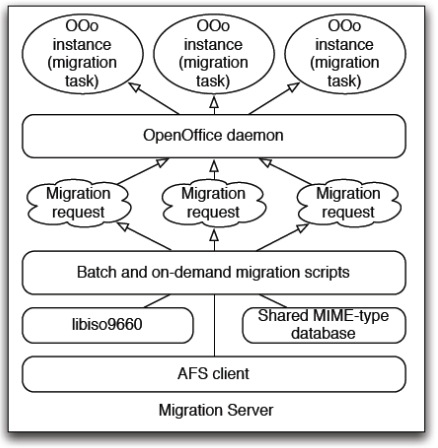

Figure 2. Migration services.

Converted documents are stored on a dedicated AFS volume. Each converted document is uniquely renamed using an MD5 hash constructed from the absolute ISO-internal path name, avoiding collisions and allowing for a simple single-directory storage path for each ISO in the collection.

Microsoft Word and PowerPoint documents are converted to Adobe PDF via a daemonized server which allocates "headless" instances of OpenOffice 2.4 to each conversion task. Conversion failures are automatically logged by the server. Additionally, the server monitors the health of each OpenOffice instance. If memory usage exceeds an administrator-defined level, or if a conversion task appears to be hung (over time measured on a sliding scale according to the size of the document), the instance is killed and the failure logged, maintaining system stability.

A similar mechanism monitors the conversion of Lotus 1-2-3 and Microsoft Excel documents to browser-friendly HTML or structured XML using Gnumeric 1.8.3 (current stable release as of writing) and appropriate filters. Microsoft Access and DBase III/III+/IV files are converted using Python modules to extract data directly from the known binary format. Additional media documents, including MPEG video and DVD video files, are migrated to flash video in a web-friendly resolution. These migration paths were selected to demonstrate access improvements in an on-demand scenario; selection of durable formats for permanent migration is necessarily more complex (Arms & Fleischhauer, 2005).

An overview of the basic dispatch model for migration tasks is provided in Figure 2. In this simplified representation, the AFS client provides access to, and retrieval of, materials held on an off-site AFS server (as shown in Figure 1). Migration requests consist of a sequence of one or more files - or a top-level directory containing the objects to be migrated - along with target format(s) and the known communications port for the OpenOffice daemon. The daemon itself is multi-threaded, and communicates with each (headless) migration instance on a unique port. The Python code is customized to a specific site installation via a simple XML configuration file. It is based on an OpenOffice daemon script provided by OpenOffice.org along with the GPL-licensed daemon distributed as part of the ERP5 Enterprise Resource Planning system (ERP5, 2008).

In previous work, we collected statistics on the distribution of file formats within the FDLP collection at the Indiana University Libraries, and provided results demonstrating the feasibility of automated migration from legacy formats (Woods & Brown, 2008). While we maintain the ability to migrate all previously examined file types, our trials in this work focused on stress testing the automated migration environment. Using the format identification procedures discussed earlier, we selected 41,403 Microsoft Word documents, 23,569 Microsoft Excel documents, and 24,780 Lotus 1-2-3 documents for batch conversion. The server successfully managed each migration task, logging conversion failures and terminating frozen instances of OpenOffice as required. The migrated materials are maintained alongside the source ISO images on the AFS.

Emulation

The current model for utilizing the FDLP materials requires physically mounting a CD-ROM on a "reference workstation." The GPO requires depository libraries to maintain such workstations for patron use and provides specifications for the required software. ISO images can similarly be mounted utilizing common tools such as "daemon tools" and hence require no fundamental change in utilization model; however, there are significant problems with the use of reference workstations that can be ameliorated through virtualization (Chang, 2006). Furthermore, the transition to a virtual CD-ROM collection offers the opportunity to eradicate the geographical barriers implicit in the current model.

While the GPO provides specifications for reference workstations `http://www.fdlp.gov/computers/rs.html`, they are updated periodically to reflect new requirements without ensuring continued access to older items. Indeed, the GPO (`http://www.fdlp.gov/computers/rsissues.html`) states that, "Libraries should also consider keeping [existing] equipment in order to access electronic products that cannot be read with newer hardware and software" (U.S. Government Printing Office, 2006).

For rarely used materials, this suggestion seems problematic. Furthermore, many of the GPO items require installation procedures that mutate the software environment in potentially incompatible ways. A natural solution to both the problem of maintaining older reference workstations and clean environments is emulation (virtualization) which requires only that hard disk images (including operating system and applications) be preserved. Standard virtualization software (e.g., VMware) can cope with libraries of such images and ensure that any mutation of the operating environment introduced by software installation can be undone.

Virtualization has the potential to simplify the preservation of reference workstations. One can imagine a pool of images shared among libraries to provide patrons of local public libraries full access to the FDLP materials. Virtualization does not solve one fundamental preservation issue: as the materials become obsolete, knowledge about how to use the required software becomes more obscure. Preserving this knowledge is a necessary step in supporting underlying technical and access issues (Heminger & Robertson, 2000).

To partially address the issue of loss of application knowledge, we have experimented with techniques to automate mounting and installing an ISO image

within a VMware virtual machine. The basic approach we are exploring is the creation of a single-install application which responds to user "click" on ISO images by selecting a local VMware machine, mounting the selected ISO image, and running an image-specific installer script. This process might be further expanded by executing specific helper applications. To determine the possible utility of this approach, we surveyed 100 ISO images containing Windows/DOS executables. Of these, more than 50 required a multi-stage installation procedure prior to use.

Because of legacy use in software testing, virtualization solutions such as VMware provide scripting interfaces enabling automated control of a virtual machine. In our work we developed a Perl script which utilizes theVMware VIX API to start and stop virtual machines as well as mount and unmount CD-ROM images[6]. We use the "snapshot" capability of VMware to ensure that all modifications made by a user are erased and to guarantee that each user is presented with a virtual machine in a known state.

The guest OS is configured with key applications for document and media browsing (Microsoft Office 97 with compatibility updates, Adobe Acrobat, VLC, and a current release ofWindows Media Player) to provide the user with a simple, easy-to-use environment for browsing legacy documents on the mounted ISO image. ISO images are mounted from the network using a standard AFS client to provide volume access to IU.EDU in the global namespace. In cases where an installation is required, a precompiled Windows executable unique to the image is copied from the AFS to the guest OS and run at startup to automate the install process. A standard "wizard" provides the user with the option to cancel the installation if desired. For this research, these executables are simple wrappers around macro-style Windows scripts, implemented with the cross-platform wxWidgets GUI library. As such, they are easily maintained and can be trivially ported to additional client platforms, should the need arise.

Our proof-of-concept trial with 100 ISO images containing legacy installation executables in the top-level directory demonstrated an additional advantage of this approach. Of the 66 images requiring local installations, the majority were hard-coded to look for a physical device such as a D: drive where the CD-ROM would originally have been mounted. This limitation is readily accommodated in an emulated environment.

Image Access

As discussed previously, we preserve CD-ROM data in the form of "bit-faithful" ISO9660 images which are supported by a well defined standard. In this section we consider three issues relating to ISO images – access to the files within the image, distribution of a shared collection of ISO images using OpenAFS, and the creation of bit-faithful images.

Access within ISO Images

ISO images are directly supported in many operating systems (BSD, Linux, OS X) and emulation tools (e.g., VMware). Furthermore, additional modules (e.g., Daemon tools) in Windows. Thus, if the only goal is preservation, ISO images are a

[6] VMware Server. http://www.vmware.com/products/server/

sufficient target. In our work, we are interested in making the CD-ROM collection more useful in virtual than physical form. This requires the ability of a server to access the contents of a large collection of ISO images.

Our first approach was to exploit the ability of Linux to mount ISO images on *loopback devices*. By suitable creation of file links and configuration of the automounter, it is possible to make a collection of ISO images appear to be mounted as subtrees of the host file system. Utilizing the Apache web server, we quickly made our collection of ISO images browsable and searchable on the web – an action which led to at least one request that embarrassing information in the public record, but previously inaccessible through Google, be removed.

There are several limitations to our first approach – many of the files are in obsolete formats, many files contain links that implicitly depend upon the ISO image being mounted as a virtual CD, some of the ISO images were created from Macintosh computers and are not fully supported by Linux, and allowing web access to trigger kernel mounting events poses scaling and possible security issues. Thus, we have moved to utilizing a widely available library, `libiso9660`, to enable direct access to the contents of ISO images without mounting.

As discussed above, some CD-ROMs created for Macintosh computers have compatibility issues. This issue is manifested in pairs of identically named files representing "resource" and "data" forks (a Macintosh concept). While the end user is generally interested in the data fork, Linux is unable to extract the correct file from ISO images – indeed we had to patch `libiso9660` to "do the right thing."

Finally, there are significant issues arising from standard ISO9660 extensions such as Joliet and Rock Ridge which were intended to overcome file naming and metadata problems in the original ISO9660 specification. These problems make it difficult to correctly render all file names and subsequently use these rendered names to find files in certain ISO images.

Image Distribution

ISO images are quite large – as much as 8 GB for recent DVD-based titles – and in most cases only a small fraction of the information in an image is required either to determine that a title is of no further interest or to satisfy a specific data query. Thus, it is extremely inefficient to download entire CD-ROM images on demand. Utilization of most CD-ROM titles is extremely low and even with rapidly declining storage costs it does not appear to make sense to mirror a large CD-ROM image collection at all libraries. Furthermore, such widespread mirroring complicates the access control required to satisfy copyright restrictions. In this section we discuss our use of existing distributed file system technology, the Andrew File System (AFS[7]) to support sharing of an ISO image collection, with proper access controls, and in a manner that largely eliminates the need to copy ISO images to satisfy patron requests. Our use of AFS supports access to CD-ROM images through web-server based browsing, through remote mounting on workstations or emulators, and copying of entire images in the rare cases where that may prove necessary.

[7] OpenAFS http://www.openafs.org/main.html

Key characteristics of AFS that we exploit are a global namespace, transparent storage migration, storage mirroring, multidomain authentication using the widely deployed Kerberos protocol, flexible access control based on access control lists (ACLs), and clients for most common operating systems. In the Emulation section we discussed a simple application that automates mounting ISO images in Vmware Workstation. Furthermore, our web application accesses both its metadata and ISO images through AFS without apparent performance problems. An exception may be high-bandwidth movies where performance is significantly improved by image copying. This level of performance stands in direct contrast to the issues observed with mounting ISO images as filesystems on a local server, including overhead on the kernel and scalability limitations.

In our prototype system, the ISO images are distributed across 5 volumes `file://afs/iu.edu/public/sudoc/volumes/[01-05]` which are accessible from anywhere by anybody. We separately maintain metadata in METS form `file://afs/iu.edu/public/sudoc/metsxml` generated from the Indiana University Libraries MARC records and which provide links to the raw images. Our web server uses Swish-E to index these METS records, and utilizes XSLT to format the METS records. We anticipate that in a production system, libraries may integrate such a collection into their own catalogs or digital repositories by mining the METS records. While all of the FDLP materials are openly available, we have created an additional collection of materials (e.g., Unesco) which are subject to copyright restrictions and where the ISO images are only available to users in the IU Kerberos domain.

The model we anticipate is one where a collective of libraries share responsibility for creation of metadata and ISO images and share these materials through a dedicated AFS domain. Individual libraries could contribute materials through a local volume server and could control access to these materials through ACLs. Access by patrons of other institutions would be supported by linking to these participating institutions' Kerberos domains and by appropriately managing ACLs. A key issue for such a collective will be the development of effective administrative policies and tools; ACLs provide an effect enforcement mechanism, but are not sufficient. For example, suppose copyright restrictions required that a particular item be accessible by only one patron at a time; while this restriction could be implemented by modifying the appropriate ACL at access time, we have not created administrative tools to perform such modifications. Additional benefits of such a distributed approach are described in a similar pilot project by Yale Library's Government Documents & Information Center (GDIC) (Gano and Linden, 2007).

Image Creation

Bit-level preservation of CD-ROMs would appear to be relatively straightforward – organization of information is governed by a well defined standard (ISO9660 (ECMA, 1987)). The data are protected by error-correcting bits which make it possible to detect and correct most errors, and many software packages exist for ripping ISO images which are standard files containing the raw data from the CD-ROM. However, our experiences in preserving more than 4,500 CD-ROM images have uncovered several significant pitfalls at both the bit-preservation and application levels. These pitfalls fall into two major categories – poor conformance to the underlying standards, and inappropriate contextual dependencies embedded in the preserved data.

Operating systems such as Linux, Unix, and Windows all treat CD-ROM drives as "block devices" in which the raw data can be accessed as a single large binary file organized in fixed-size blocks. For CD-ROMs, these blocks are called sectors and typically consist of 2,048 bytes of data with additional error-correcting bits used by the drive hardware to detect and correct bit errors.

The operating system interprets the contents of this binary file to provide *a file system* view consisting of a tree-shaped hierarchy of directories and files which can be accessed by applications through standard file operations such as *open, read, write*. The binary file is organized according to the ISO9660 standard (ECMA, 1987) (described below) – In principle, preserving the contents of a CD-ROM consists of copying this binary file (called an ISO image) onto another medium. For example, Microsoft provide instructions for doing just this[8] and most available Windows tools for creating ISO images appear to follow this basic procedure.

There are two significant problems with simply copying the bits off a CD-ROM: there is no obvious way to know that you have all the bits; and there is no way to know the bits that you have are all correct. The latter problem is ameliorated by the error correction bits on the CD-ROM which are utilized by the CD-ROM drive to detect and correct errors; however, for an archival copy this may not be sufficient. The problem of knowing whether you have all the bits is complicated by the fact that CD-ROMs are typically created with additional blocks of zeros to assist in the physical process of extracting the bits that are part of the file system. Errors in reading these extra blocks are irrelevant. As we shall show, knowing that you have all the bits requires interpreting the underlying file system organization.

As mentioned, the file system on CD-ROMs is organized according to the ISO9660 standard. An ISO file system consists fixed-size *sectors* organized in one or more volumes. Each volume begins with a dedicated sector, called a volume descriptor. This volume descriptor includes fundamental information such as an identifier, volume size (in sectors), sector size, and pointers to directory information (within the volume). The directory information includes *path tables* – a largely obsolete mechanism for quickly finding files, and a root directory. As with most file systems, directories are implemented as binary data structures embedded in ordinary files.

One approach to determining the amount of data within an ISO image is to find the volumes, and compute the length of the volume from the volume header. This is greatly simplified by the fact that most published CD-ROMs (all in our data set) consist of a single volume. Unfortunately, the volume header information is frequently wrong – 19% of the CD-ROM images we created had incorrect size information in their headers. The number can be both too high or too low. One fairly benign case arises in "track at once" recording when images are one sector shorter than advertised. The advertised length can also be too high when the recording software computes the image size prior to "compacting" the file system. This problem emerged in our work when we began experiments with file migration and found significant numbers of truncated files within the ISO images we had created.

To circumvent the "bad header" problem, we wrote programs that walk the file

[8] Microsoft Help and Support: How Win32-Based Applications Read CD-ROM Sectors in Windows NT http://support.microsoft.com/kb/138434

system in an ISO image computing the starting sector and length of each file (including directories). Using this technique we were able to determine the "true" end of the image[9]. Unfortunately, approximately 10% of the images we created with Windows-based software were truncated before the end of the image. Experiments with a variety of Windows-based tools on multiple machines confirmed this behavior. In contrast, the Linux tool `dd` supports copying all of the raw data from a CD-ROM. In an experiment with 86 CD-ROMs whose images were truncated by Windows, we were able to read all of the ISO image for 81 using dd. Of the remainder, one CD-ROM was cracked. Thus, we expect the rate of failure to read all the bits of CD-ROMs to be under 1%, provided the right tools are used.

Once it has been established that an ISO image contains all the relevant bits, it remains to determine if these bits are correct. Since the CD-ROM publications of the GPO provide no additional checksum information, the only viable approach is to compare at least two images created from different copies of a CD-ROM title for consistency. As discussed above, it is crucial that only the relevant bits in an image be compared (or checksummed) as there is significant potential for spurious errors. In the case of the GPO publications, comparing copies of CD-ROMS is further complicated by the use of different identification schemes in the various FDLP libraries (the SUDOC number system is not universally applied and poses significant potential for ambiguity). Thus, simply identifying two copies of the same publication may require significant effort. One strategy we have explored is generating checksums for the first 1 Mb of each image in our collection as a convenient hash value for determining whether two CD-ROMs are likely to be the same publication.

Discussion

The work presented here addresses fundamental access problems faced by institutions with legacy CD-ROM holdings. Our project complements and operates alongside existing frameworks without significant additional overhead. It uses interoperable metadata and low-cost open source tools, and further supports secure, flexible sharing of archival materials. These factors, along with viable strategies for file format identification, flexible migration profiles, and emulation support for legacy environments, provide a blueprint for future success in handling these types of collections.

Systems such as Fedora (Petinot et al., 2004), Greenstone[10] (Witten et al., 2001) and DSpace[11] provide an established basis for archival management systems. In our view, there are fundamental access and preservation issues with CD-ROM collections that these systems do not adequately address. Foremost is the fact that CD-ROM collections typically consist of a comparatively small number of large objects (physical CD-ROMS and DVDs, or their bit-identical ISO-9660 images) that will generally see only fractional access. While the images as a whole contain a large number of disparate file types, these files are bound within the context of individual ISO images. Our approach emphasizes the inherent interrelationship of items within an ISO image over those between images.

[9] This works for interchange level 1 and 2 CD-ROMs because they require all files to be contiguous. Interchange level 3 appears to be rare since it is incompatible with most operating systems.
[10] Greenstone Digital Library Software http://www.greenstone.org
[11] DSpace http://www.dspace.org

This project explicitly enables libraries to build shared virtual collections using predominantly off-the-shelf, open source tools. The methods discussed in this paper address a number of outstanding access issues faced by institutions holding legacy digital materials, and may be readily integrated into existing infrastructure.

References

Arms, C., & Fleischhauer, C. (2005). Digital formats: Factors for sustainability, functionality, and quality. In *Archiving 2005*. Society for Imaging Technlogy.

Chang, W. (2006). NIST data preservation and migration strategy: Virtualization. Retrieved November 12, 2006, from http://www.itl.nist.gov/iad/894.05/gipwog/Feb-2-06/NIST_DPMTestBed_at_GPO.pdf

Council on Library and Information Resources. (2000). *Authenticity in a digital environment*. Technical report, Council on Library and Information Resources: Washington D.C. Retrieved October 14, 2009 from http://www.clir.org/pubs/reports/pub92/pub92.PDF

Depository Library Council. (2006). *Knowledge will forever govern. A vision statement for federal depository libraries in the 21st century*. Retrieved October 14, 2009, from http://www.access.gpo.gov/su_docs/fdlp/council/index.html

ECMA. (1987). Standard ECMA-119:Volume and file structure of CDROM for information exchange, 2nd edition (December, 1987). Retrieved October 8, 2009, from http://www.ecma-international.org/publications/standards/Ecma-119.htm

ERP5 (2008). How To Use OOOD. Retrieved October 14, 2009, from http://www.erp5.org/HowToUseOood

Gano, G. and Linden, J. (2007). Government Information in Legacy Formats: Scaling a Pilot Project to Enable Long-Term Access. Retrieved October 14, 2009 from http://www.dlib.org/dlib/july07/linden/07linden.html

Heminger, A. R., & Robertson, S. (2000). The digital rosetta stone: A model for maintaining long-term access to static digital documents. *Communications of AIS 3*(1es):2.

Indiana State Library (2006). Federal documents depository program. Retrieved October 14, 2009, from http://www.in.gov/library/feddeposit.htm

Leonard, T. (2008). The shared MIME-info database standard. Retrieved October 14, 2009, from http://standards.freedesktop.org/shared-mime-info-spec/shared-mime-info-spec-latest.html

Maniatis, P., Roussopoulos, M., Giuli, T., Rosenthal, D., & Baker, M. (2005). The LOCKSS peer-to-peer digital preservation system. In *ACM Transactions on Computer Systems (TOCS), 23*(1). Retrieved October 13, 2009, from http://portal.acm.org/citation.cfm?id=1047917

Mellor, P., Wheatley, P., & Sergeant, D. M. (2002). Migration on request, a practical technique for preservation. In *ECDL '02: Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, 516–526. Springer-Verlag: London.

MTR (2005). Minimum Technical Requirements for Public Access Workstations in Federal Depository Libaries. Retrieved October 14, 2009, from http://www.fdlp.gov/administration/computers/244-mtr

Petinot, Y., Giles, C. L., Bhatnagar, V., Teregowda, P. B., Han, H., & Council, I. G. (2004). A service-oriented architecture for digital libraries. *Proceedings of the Second International Conference on Service Oriented Computing, New York City*.

Rabinowitz, J. (2004). Simple Web Indexing for Humans (Enhanced). Retrieved October 14, 2009, from http://www.swish-e.org

U.S. Government Printing Office. (2006). Depository library public service guidelines for government information in electronic formats. Retrieved November 30, 2006, from http://www.access.gpo.gov/su_docs/fdlp/mgt/pseguide.html

Witten, I. H.,Bainbridge, D., & Boddie, S. J. (2001). Power to the people: End-user building of digital library collections. In *Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries*, 94–103. ACM Press.

Woods, K., & Brown, G. (2008). Migration performance for legacy data access. *International Journal of Digital Curation 3*(2), pp. 74-88.