Long-Term Preservation of Digital Records: A Technical Solution
Henry M Gladney
Abstract

The proper entities for archival attention are patterns inherent in transmitted and stored messages. Most digital archival repository technology—what private sector enterprises call content management (CM) technology—has been thoroughly understood and widely deployed for more than a decade. This technology is not adequate for long-term digital preservation because it includes no mechanisms for reliably assuring authenticity and intelligibility of digital documents for fifty years or longer. CM provides for near-term preservation without handling long-term preservation, which must overcome risks associated with technological obsolescence and fading human memory. We offer a solution to mitigate these risks. Implementing the needed software would be a small addition to widely deployed CM offerings. We show that our long-term preservation solution, devised for cultural and scholarly digital documents, is already structured to support archival principles for business records, and we describe this Trustworthy Digital Object (TDO) architecture and its design core sufficiently to show how archivists can participate in managing digital repositories that conform and are attuned to the particular needs of any archival institution.

Introduction

> Without a systematic and significant effort to develop tools and techniques that substantially mitigate the consequences of limited media life expectancy and hardware and software obsolescence, . . . we risk substantial practical loss, as well as the condemnation of our progeny for thoughtlessly consigning to oblivion a unique historical legacy.[1]
>
> Preservation of digital information is not so much about protecting physical objects as about specifying the creation and maintenance of intangible electronic files whose intellectual integrity is their primary characteristic. Preservation in the digital world is not exclusively a matter of longevity of…fragile storage media.…The viability of digitized files is much more dependent on the life expectancy of the access system.[2]

For challenges created by new technology, solutions usually include additional new technology. Digital preservation work has been directed primarily at scholarly and cultural information.[3] *Information* is the subset of knowledge that a human being can communicate to

---

1 Charles M. Dollar, *Authentic Electronic Records: Strategies for Long-Term Access* (Chicago: Cohassett Associates, 2000): Introduction p.3.

2 Society of American Archivists, *The Preservation of Digitized Reproductions* (1997), available at http://www.archivists.org/statements/digitize.asp, accessed 20 April 2009.

3 Christopher A. Lee and Helen R. Tibbo, "Digital Curation and Trusted Repositories: Steps Toward Success," *J. Digital Information* 8, no. 2, 2007. For a definition of digital preservation, see Michèle V. Cloonan and Shelby Sanett, "Preservation Strategies for Electronic Records: Where We Are Now—Obliquity and Squint?," *American Archivist* 65 (Spring/Summer 2002): 95. For discussion of authenticity, see Maria Guercio, "Principles, Methods, and Instruments for the Creation, Preservation, and Use of Archival Records in the Digital Environment," *American Archivist* 64 (Fall/Winter 2001): 251.

another human being by speaking, writing, or drawing. In contrast, *knowledge* is what a human being or animal can know, including that of which he or she might not be consciously aware, as Sigmund Freud taught, or be unable to communicate in words, such as how to ride a bicycle. The emphasis and language in discussion of archiving organizational records are so different from those about preserving cultural works that readers might think different methodologies are needed. For instance,

> Unlike other types of information objects…records are created within a universe of discourse where there is often a high degree of shared information and expectations among participants.…In such contexts, important information is often conveyed by form, as well as by substance.…[P]articipants expect certain forms to be used for certain types of transactions...Common knowledge…provides a systemic check…on the reliability of their records.…To enable parties who were not participants in a process to understand the records of that activity,…an archival system should contain and convey information about the types of records typically produced, the elements of intrinsic and extrinsic form of each type, the relationships between processes and records, and also the implied knowledge…common to participants.[4]

We find that the long-term preservation measures needed for scholarly information are also sufficient for archival records, since these measures were designed to avoid aspects that distinguish different kinds of information.

Long-term digital preservation (LDP), the complex of measures required for and/or undertaken to mitigate digital object unreliability caused by ravages of time, including human misfeasance, fading human memory, and technological obsolescence, is best thought of as an extension of near-term content management (CM) services. *Content management* is a twenty-first-century commercial name for the complex of services required to preserve, protect, and make accessible information mostly created by people other than its custodians. *Content management* grew out of what in the 1990s was called *digital library* services. To influence archival technology and to control how it is applied in their institutions, archivists need a high-level understanding of CM software offerings, especially those aspects that repository managers can control. This paper sketches the structure that we believe must be added to current CM technology to make information reliably useful many years from now. We sketch a mechanism that makes any record's authenticity, or lack thereof, readily testable by anybody who depends on that record.

The core of this extension is a way of structuring information objects to contain or to refer reliably to context that their creators and keepers believe essential and a way of representing such objects so that our descendants will be able to interpret them correctly. We call this core *Trustworthy Digital Object* (TDO) methodology and prescribe a formal structure and usage for TDOs.[5] This paper outlines the main aspects of this approach, hopefully describing it enough so that archivists can collaborate effectively with software engineers in achieving archival goals.

---

[4] Kenneth Thibodeau, *Overview of Technological Approaches to Digital Preservation…*, in CLIR, *The State of Digital Preservation: An International Perspective,* Conference Proceedings (2002), available at http://www.clir.org/pubs/reports/pub107/thibodeau.html, accessed 23 April 2009.

[5] H. M. Gladney, *Preserving Digital Information* (Heidelberg: Springer Verlag, 2007) cites hundreds of background articles. Also see the author's website at http://home.pacbell.net/hgladney, available on 12 May 2009.

Synopsis and Scope

> There is a clear sense of the broad requirements for protecting authentic electronic records over time. What is missing is a mechanism for translating this sense into rigorous procedures that can be adapted and incorporated into auditing and accounting guidelines for authentic electronic records. [6]

The proper entities for archival attention are patterns inherent in transmitted and stored messages. In this article, a *message* is a brief information string that is to be conveyed from some *writer* to some eventual *reader*. To be considered authoritative and of interest for preservation, the representation of such a pattern must be truthfully and firmly bound to adequate provenance information. We outline conventional digital repository architecture and TDO methodology, providing enough technical description so that readers can understand the services possible and how these can be made robust enough for useful information access in the distant future, even though nobody knows how technology will evolve. We refer readers who might want to know details of these services to technical descriptions. The word *reader* denotes a role depicted in Figure 1, being somebody who uses information. The locution is used to suggest independence of the content, style, or purpose of the information under discussion, e.g., it might be is a computer program to be executed. The reader might not be a human being, but a machine process instead.

We find three models particularly helpful for describing digital preservation technology: first, a model of message transmission between creators and users, depicted in Figure 1; second, a model of digital content repository infrastructure, depicted in Figure 2; and third, a model of annotated digital content, suggested in Figure 4. A full version of each figure is included for completeness, even though this paper explains only a subset of its features. Our review is limited to technical aspects of long-term digital preservation, leaving to other authors topics such as selecting what to save[7] and assisting human managers of repository institutions to plan their work and manage collections.[8]

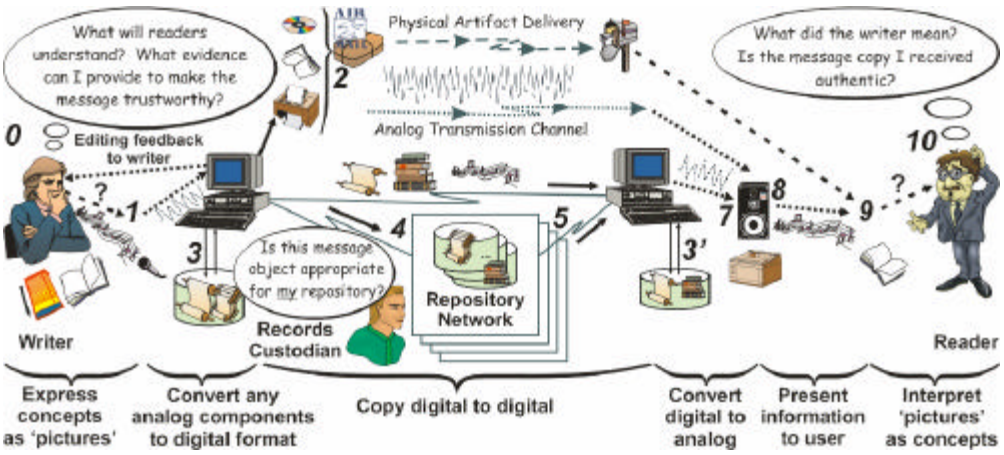

Figure 1: Depiction of a model for documentary information communication.

---

6 Dollar, *Authentic Electronic Records*, §4.3

7 Cloonan and Sanett, "Preservation Strategies."

8 For instance, see the website of the EU *Preservation and Long-term Access through Networked Services* (Planets) project at http://www.planets-project.eu, accessed 23 April 2009, and work it cites.

What Would an LDP Solution Accomplish?

> In both the library world and the world of archives, people at times have become so focused on the artifacts themselves that they have risked losing sight of their users or their users' needs. A similar tendency exists in the world of computers—the tendency to turn inward and become preoccupied with the computational artifacts, with their elegance, simplicity, internal consistency and so on....
>
> The challenge ahead is to bring our best technical skills to bear on the problem of digital preservation without losing sight of the ultimate human purposes these efforts serve, purposes which cannot be found inside the machines we are busy programming or using.[9]

What requirements would an LDP solution address? What might our descendants expect of information stored today? They would be satisfied if, for whatever record is of interest, they could:

1. Retrieve a copy of the bit-strings that represents the content if authorized to do so. A *bit-string* is a sequence of bits that represents information. This word, which in many contexts is a synonym for *file*, is used to suggest that no layout details are relevant. In contrast, a computer *file* is a representation partitioned into chunks that are conveniently laid out and managed on a magnetic or optical storage disk or tape.
2. Read or otherwise use the content as its creators intended, without adverse effects caused by mistakes and inappropriate changes made by third parties.
3. Decide whether the information received is sufficiently trustworthy for their application. Archivists interested in diplomatics call this the ability "to distinguish the false document from the true one."[10]
4. Exploit embedded references reliably to identify and retrieve contextual information and to validate the trustworthiness of contextual links, doing so recursively to as much depth as is wanted.
5. Exercise all this functionality without hindrance by technical complexity that could be hidden.

In addition to authors, editors, archivists, and librarians, some citizens will want to preserve information without needing to ask anybody's permission to do so. They will want convenient tools and infrastructure to:

6. Package any content to be LDP-ready, doing so in some way that ensures that their descendants can use this content as specified above.
7. Submit such readied content to repositories that promise to save it reliably, possibly in return for a fee for archiving service. (People are willing, in anticipation of death, to pay for storing their body remains. We believe they can be persuaded to pay for storing their intellectual remains together with high-quality provenance information.)

What technology will repository institutions want? In addition to perfect-world digital library technology, they will want support for:

8. Continuing to use their currently deployed repository software without disruption originating in extensions for LDP; replacing parts of this software in future years to provide their clients with the best services, doing so without disturbing already

---

[9] David M. Levy, "Heroic Measures: Reflections on the Possibility and Purpose of Digital Preservation," *Proceedings of the Third ACM Conference on Digital Libraries* (1998), 152–61, especially page 159. ??

[10] Heather MacNeil, *Trusting Records: Legal, Historical, and Diplomatic Perspectives* (New York: Kluwer Academic, 2000), §4.1.

preserved information. A *client* is someone who benefits from digital repository services, either by storing records for other clients or by reading such stored records; alternatively, in *client-server computing*, a machine that requests services from a remote machine.

9. Sharing preserved content and metadata without adjustments requiring human judgment. *Metadata* convey structured information associated with a digital content object, often provided by people other than the content author(s) to make it easier to use and manage that content, often conforming to EDP standards.
10. Sharing preservation effort with their clients to avoid burdens beyond repository resources.
11. Ensuring that preserved information survives the demise of a large subset of all repositories.

Human users will want every step to be as automated as it can be without interfering with their subjective choices. Eliminating the distraction of clerical tasks will free them to focus on those activities that only human beings can accomplish, such as creating, organizing, selecting, and preserving.

Communication Is Complicated

Written human communication is complicated, partly because we are sensitive to syntactic nuances of messages and partly because some message ambiguity seems to be unavoidable. Among digital records, the consequences are particularly obvious in unstructured text documents and in editing programs that we use to create, alter, and inspect text. The vendors of tools such as Microsoft Office compete with small armies of programmers who construct and maintain their editing programs. For instance, the Microsoft team made a great effort to ensure that documents prepared with its Microsoft Word 2007 offering can be edited with its earlier Microsoft Word 2003 offering. We sense how complicated text documents are, and how sensitive we are to nuances, when we convert files from/to Word format to/from the Sun Microsystems OpenOffice Writer format and are annoyed by small discrepancies when we compare printed outputs.

This intrinsic complexity cannot be avoided in digital preservation.

Technical Requirements

In the current article, *long-term* pertains to aspects wanted by future users who might have questions when today's creators and custodians are no longer available to answer. *Near-term* describes measures undertaken to meet the needs of users today and in the next five to ten years—a period short enough for repository managers to ascertain client satisfaction and to react with service improvements.

A comprehensive treatment for long-term information protection needs to consider all sources of unreliability, matters of scale, and integration with extant CM infrastructure.[11] These measures address all aspects of ingestion into repositories, curation, cataloging, access provision and business controls, and storage management. Nevertheless, we assume that near-term CM is a different, albeit related, topic that is already well handled in hundreds of articles. *Digital curation* is a librarians' term for management of digital objects over their entire lifecycle,

[11] Steen S. Christensen, *Archival Data Format Requirements*, report from the Denmark Royal Library (2004) available at http://netarkivet.dk/publikationer/Archival_format_requirements-2004.pdf, accessed 12 May 2009.

ranging from pre-creation activities wherein systems are designed, and file formats and other data creation standards are established, through ongoing capture of evolving contextual information for digital assets housed in archival repositories.

Since software developers will want LDP technology to integrate smoothly with near-term CM services, we will now discuss some aspects of near-term CM services to lay groundwork for our assertion that the extra software for LDP can be a surprisingly small addition to current CM software. Two questions are prominent:

1. Is it feasible to represent and package digital content to accomplish such a strategy?
2. And, if so, precisely how can this be done?

Table 1. Generic Risks to Digital Records

Generic risk	Examples
Media and hardware failures	Failure causes include random bit errors and recording track blemishes, breakdown of embedded electronic components, burn-out, and misplaced offline disks and tapes.
Software failures	Most practical software has design and implementation deficiencies that might distort communicated data.
Communication channel errors	Failures include detected errors (one packet error in 10 million instances) that are automatically corrected and undetected errors (bit rate of ~10^{-10}), and also network deliveries that do not complete within a specified time interval.
Network service failures	Information accessibility might be lost from failures in name resolution, misplaced directories, and administrative lapses.
Component obsolescence	Media, hardware, or software components might become incompatible with other system components within a decade of their appearance. File format obsolescence might prevent content decoding and rendering.
Operator errors	Human operator actions in handling any system component might introduce irrecoverable errors, particularly at times of stress during execution of system recovery tasks.
Natural disaster	Floods, fires, and earthquakes.
External attacks	Deliberate information destruction or corruption by network attacks, terrorism, or war.
Internal attacks	Misfeasance by employees and other insiders for fraud, revenge, or malicious amusement.
Economic and organization failures	A repository institution might become unable to afford its computer running costs or might vanish entirely, perhaps through bankruptcy or mission change, so that preserved information suddenly is irrelevant to its sponsors and custodial care is abandoned.

Generic Risks and Engineering Considerations

General risk sources are suggested in Table 1.[12] Which of these risks are important will, of course, depend on the information genre and source of each record. Many of these risks are effectively handled by distributed file replication, with errors detected by cyclic redundancy checks (CRC), each of which is a fixed-length function of a variable-length bit-string used to

[12] Adapted from David S. H. Rosenthal, Thomas S. Robertson, Tom Lipkis, Vicky Reich, and Seth Morabito, "Requirements for Digital Preservation Systems: A Bottom-up Approach," *D-Lib Magazine* 11, no. 11 (2005).

detect data errors during transmission or storage. CRCs are simple to implement, easy to analyze mathematically, and excellent for detecting common reliability failures—methods that have long been standard practice and therefore need little attention here.

The short life (five to ten years) of magnetic and optical media has recently stimulated investigations of the practicality of digital recording on microfilm.[13] Audio-video bit-strings can be interspersed with human-legible short text sections. Since preserving bit-string copies indefinitely is well understood, our discussion is limited to showing how it fits into CM architecture.

> While paper-based materials require all-the-time care, benign neglect is not always harmful. In fact, in some instances it is better *not* to treat an item. Digital materials, on the other hand, require constant refreshing, reformatting, migrating, etc. These represent much more pro-active and costly endeavors. For digital materials, neglect may result in total loss.[14]

This quotation illustrates how tricky and misleading economic comparisons between documents on paper and digital documents can be. In addition to its flawed assertion about reformatting, this statement's "costly" fails to say whether its intended comparison is for total repository costs or for costs per record. Nor does this discussion include the largest cost of preserving information on paper or other analog documents—the production process. Arguably, the corresponding digital cost includes preparation for preservation, a step seldom taken currently and that, when it has been reduced to routine practice, is likely to be only a small addition to the cost of producing each record and smaller than the cost of creating a paper record. We intend to show how to develop technology for low-cost digital preservation—software ready for deployment and institutionalization.

Developmental Context

A comprehensive CM specification would be lengthy. Each of many infrastructure elements is not only sophisticated, but also the focus of rapid, highly competitive improvement. For example, content search tools are much more powerful today than they were only five years ago and will surely continue to be improved. Each human user, and also each repository, is likely to want as much freedom and autonomy as possible without service limitations. Digital content creators, users, and custodians want autonomy, but so do software developers and information curators.

Software providers typically compete with focused services such as search tools, extraction of search indexes from core content and metadata, database technology for repository catalogs, file management tools, cryptographic data protection tools, and edit programs. Although we cannot predict how any infrastructure element will evolve, we are confident that today's important capabilities will be reproduced in every future computing infrastructure and that software providers will ensure migration to comply with evolving interface standards. What is likely to interest archivists is not how service quality can be maximized, but instead what opportunities are available for tailoring services to their specific institutional preferences.

Software Architecture

---

[13] Norbert Bolewski, *Langzeitarchivierung über "Bits on Film"* (2008), available at http://fkt.schiele-schoen.de/a12394/Langzeitarchivierung_ueber_Bits_on_Film.html, accessed 12 May 2009.

[14] Cloonan and Sanett, "Preservation Strategies," 85.

> Archival perspectives and planning need to be built into the creation and early management of all information that will enjoy long-term preservation. Ironically, in this process of extending the archivist's influence and control, there is a concomitant loss of control. Simply put, if creators of digital information do not take steps to preserve it early in its life, it will never reach any long-term preservation facility. This recognition leads to the need for concepts of responsible custody and archiving to pervade society.[15]

Helen Tibbo's expectations have a profound implication for LDP strategy. The pace and extent of content management enhancements are so great that the most successful approach is likely to be one that stays clear of this juggernaut to avoid being overwhelmed. Prudence suggests that LDP software should not interfere with infrastructure components such as those suggested below. It further seems prudent to avoid incorporating any such component into LDP software because doing so would risk rapid obsolescence of that software.

Our LDP solution achieves generality partly by separating long-term preservation from near-term archiving tools as much as possible. This makes it compatible with most deployed CM software, allowing repositories to adopt LDP support without disrupting current operations.

Architectural Strategy

In our preservation architecture we attempted simplicity and economy, doing so by iterative refinement with particular attention to well-known software engineering techniques. We began with a keen sense of what will be expensive and what will be relatively inexpensive. Human time and efforts are expensive, ever more so relative to the costs of storage space and computing cycles. As the number of records that people want to preserve increases, it will become cost effective to automate any human work that could be done by a machine. Effort to automate whatever can be must start by clear thinking about the boundary between necessarily subjective decisions and objective specifications.[16]

The strategy continues with identifying work done by others and technical aspects that are already, or will soon be, stable. We focus on a structure for combining other developers' work—their components and their interfaces, separating thinking about what should be in any component from thinking about integrative structure. We choose to focus on the most difficult preservation challenges. For example, for ensuring enduring information trustworthiness, we focus on the records whose improper modification is the most tempting opportunity for fraud and other mischief. For information intelligibility, we focus on computer programs and rules in which tiny changes might lead to disastrous behavior, such as sending a space traveler to the sun instead of the moon.

If such difficult cases can be handled by general mechanisms that are also simple and economical, as proves to be true for trustworthiness, then all cases will have been handled. If not, as proves to be the case for intelligibility, simpler or more economical methods can be introduced as "plug-ins" for digital object types that occur sufficiently frequently for special treatment to be affordable.

---

[15] Helen Tibbo, "On the Nature and Importance of Archiving in the Digital Age," *Advances in Computers* 57 (2003): §7.6, 2–69.

[16] Gladney, *Preserving Digital Information*, §3.3.

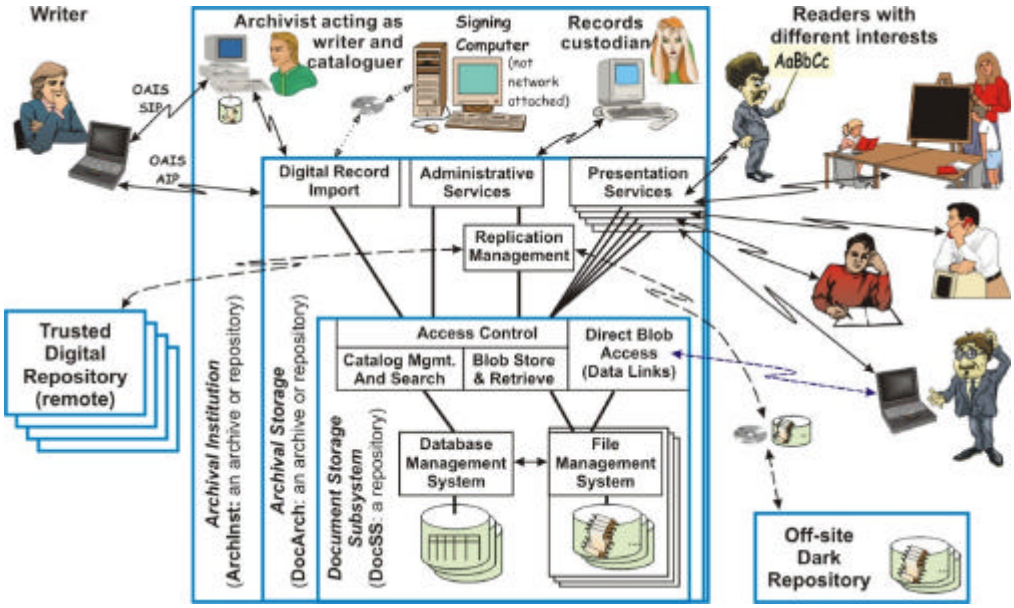

Figure 2: Depiction of a model for repositories, human roles, and context
The diagram is a representation familiar to software engineers. Each box suggests its **purpose** without specifying technological details of **how** the called-for result is to be accomplished. Specifying the kinds of data allowed to pass between the boxes is a central aspect of an architecture.

Repository Infrastructure

In digital repositories, each object is registered with a unique identifier, a bit-string copy is stored in a safe place, and entity descriptors are factored into a catalog database that also points at the stored bit-string. Figure 2 depicts a model helpful for discussing repository management.[17] Critical archiving aspects are immediately apparent. For instance, all human interactions are via personal computers. Thus, most data preparation for preservation can be accomplished using available content and metadata editing programs, with no more than modest additions to make records durable for the long term. The figure shows nested repositories.

An archival institution, ArchInst, tries to address everything pertinent to the safety and ready availability of records within its mission, whether those of the parent institution or those collected from other people or organizations. Within ArchInst, a computing complex, DocArch, presents numerous user interfaces to ArchInst employees and their clients—or, more precisely, to a mechanical agent for each such human being. It also presents interfaces to other repositories that might be organizationally autonomous.

DocArch contains a document storage subsystem, DocSS. Many different DocArch designs might accommodate potentially large differences among archival institutions; for instance, small institutions are likely to have different needs than national archives. They might also require extensive customization of parameters for institutional preferences and for coupling to document manipulation tools. In contrast, the software implementing DocSS, which provides file storage and catalog integrity management services, could be the same for every archival institution, no matter what specialized services and customization its clients might want.

---

[17] The diagram is a representation familiar to software engineers. Each box suggests its purpose without specifying technological details of how the called-for result is to be accomplished. Specifying the kinds of data allowed to pass between the boxes is a central aspect of an architecture.

The model in Figure 2 might be compared with the much-cited Open Archival Information System (OAIS) model,[18] which emphasizes different repository aspects. Specifically, OAIS emphasizes human administrative roles within an ArchInst and what it means to be a repository, providing vocabulary for discussing digital preservation.

Data security is a critical issue and a complex topic. In addition to day-to-day concerns described in the popular press and business periodicals, special concerns arise for long-term preservation. Records must be protected against improper alteration, possibly even by misbehavior of archives employees, over a very long period. Digital signatures must be validated many years after they were created. Both challenges are exacerbated by network hackers, including individuals who attack merely to demonstrate their cleverness.

In addition to widely discussed security measures, less common ones seem prudent—isolation of critical data from the Internet and data replication in autonomous repositories. Two kinds of data are critical: private cryptographic keys used to construct digital signatures and codes for bit-string integrity testing, and safety copies of whatever records might be targets for Internet criminals. Such information can be protected well by storing a copy in a computing complex that is never attached to any computer network; these data are used only to create digital signatures and to check suspected alteration of working records, and they can be accessed only by repository custodians using methods thought to be almost impossible to bypass.

How these tools connect is suggested by Figure 2, in which the Signing Computer depicts protection by isolation for secret cryptographic keys needed to create digital signatures. The Off-site Dark Repository connection depicts isolation of copies. In each case, data are copied to mountable storage volumes, such as optical disks, that are moved between the networked repository service environment and computing machines that are never made accessible from the Internet. For a digital repository, *dark* means deliberately unavailable for serving ordinary clients. For instance, my personal dark repository is copied to a set of DVDs held in a bank vault. In view of today's intense battle between "black hats" and "white knights," care is required that no Trojan horse programs accompany the transferred data into the protected machines. More generally, the technical and administrative procedures to accomplish dark repository safety deserve a level of attention that they have not yet received, except perhaps in clandestine military and intelligence organizations.[19]

Replication between active repositories, pioneered for digital preservation at Stanford University,[20] sees considerable use and is substantially similar in independent implementations. The replicating repositories should be remote from the base repository and autonomous. From the local DocSS perspective, a replicating repository looks like an ordinary user, except that its access control constraints might be somewhat different than those for human users.

DocArch ideas seem mature, but implementations are not. Development of DocArch, both user interfaces and storage systems, takes place both in academic institutions and in for-profit businesses. Refereed academic and professional publications report the first.[21] In contrast,

---

[18] CCSDS, *Reference Model for an Open Archival Information System* (OAIS) (2001), especially its Figure 4-1 on page 4-1, available at http://public.ccsds.org/publications/archive/650x0b1.pdf, accessed 12 May 2009.

[19] This topic is an enhancement candidate for the Repository Audit and Certification (TRAC) initiative; see http://wiki.digitalrepositoryauditandcertification.org/bin/view, accessed 12 May 2009.

[20] Vicky Reich and David S. H. Rosenthal, "LOCKSS: A Permanent Web Publishing and Access System," *D-Lib Magazine* (June 2001). (LOCKSS is an acronym for Lots of Copies Keep Stuff Safe.)

[21] Examples are the *ACM Transactions on Information Systems* and the online *Journal of Digital Information*, available at http://journals.tdl.org/jodi, accessed 12 May 2009.

trade press periodicals[22] describe commercial offerings[23] with different language and style. Neither literature makes much reference to the other. No careful comparisons between academic and commercial CM offerings seem to have appeared.

The DocSS repository core was originally devised to ensure catalog? collection consistency, such as ensuring that no catalog record includes a dangling reference.[24] It also hides lower-level configuration and technology details—particularly changes to exploit the latest storage technologies. From a research perspective, this component is mature, with the beginning of a standard interface definition[25] and many commercial and open-source implementations. Ongoing refinement addresses performance enhancement, flexibility, reliability, security, and interfacing the latest storage devices.

We speculate that it will gradually become easy, if it is not already, to assemble repository software from components provided by commercial and open-source suppliers to create functional replacements for offerings such as iRODS,[26] Fedora,[27] DSpace,[28] and Greenstone[29]—replacements that are easily tailored to institutional circumstances. When such component offerings conform to interface standards, such as XAM[30] and JSR 170,[25] it will be easy to replace one offering by another to exploit emerging technology implementations, doing so without disrupting deployed repository services.

Any of several DocArch components might include the administrative structure suggested by Figure 3, as might any of the digital record import, replication management, presentation management, and access control components suggested by Figure 2. Such an administrative control service allows a repository administrator to specify institutional policy rules, with any rule potentially sensitive to the identity of the human client being served, the type of data being communicated, or any of many other circumstances of the action at hand. Current research is directed toward refining rule languages[31] and providing interactive interfaces so that custodians without high EDP (electronic data processing) expertise can manage repositories.

---

22 See weekly publications such as *Information Week* and *eWeek*. [add URLs]

23 For instance, see Hitachi Data Systems, *Active Archive: A Blueprint for Long-term Preservation of Business-Critical Digital Data* (2008), available at http://www.hds.com/assets/pdf/sb-active-archive.pdf, accessed 12 May 2009.

24 H. M. Gladney, "A Storage Subsystem for Image and Records Management," *IBM Systems Journal* 32, no. 3 (1993): 512–40.

25 Java Community Process, *JSR 170: Content Repository for Java Technology API* (2006), available at http://jcp.org/en/jsr/detail?id=170, accessed 12 May 2009.

26 See San Diego Supercomputer Center information at https://www.irods.org/index.php/Documentation, accessed 12 May 2009.

27 Cornell Digital Library Research Group information at http://www.fedora-commons.org/, accessed 12-May 2009.

28 This software is used to promulgate MIT research articles. See http://dspace.mit.edu/ and http:/www.dspace.org/, accessed 12 May 2009.

29 University of Waikato software available at http://www.greenstone.org/, accessed 12 May 2009.

30 See http://www.snia.org/forums/xam/technology/standards/. An XAM review is found at http://www.informationweek.com/news/storage/showArticle.jhtml?articleID=209903838, both accessed 12 May 2009.

31 A "rule language" expresses what is wanted as assertions such as "each metadata object must contain the date at which the described object was created," instead of as a more conventional program prescribing how to implement the rule. For instance, see the RuleML description at http://www.ruleml.org, accessed 22 April 2009.

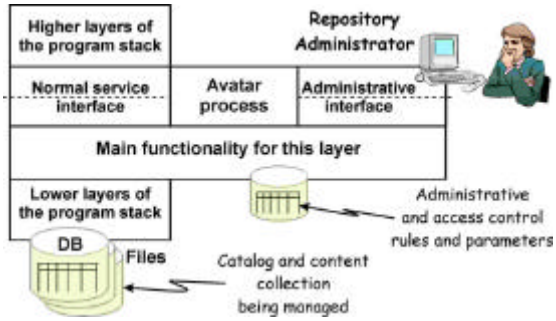

Figure 3: Model of administrative control services

As Figure 3 suggests, any service component might include an avatar process in addition to the normal service and administrative service processes. Such an avatar, an autonomous automatic process that accomplishes clerical tasks, following rules stored in its administrative database, might spring into action whenever some preplanned event occurs. Such an event could be the expiration of a timer, some condition within any accessible storage, receipt of some external signal, or any combination of such circumstances. The possibilities are limited only by the information available in the databases, the quality of the rules language, and the imagination of managers and software engineers. Experience with existing business systems shows that rule changes can be introduced without disrupting service to clients.

It is unlikely that catalogs for library and archival holdings can be designed today to be fully satisfactory for many years. Modern relational database technology has been designed to accommodate change.[32] Database extensions and rearrangements can be made without disrupting ongoing service. Scores of research papers annually explore service improvements responding to shifting institutional interests, new data formats coming from unforeseen sources, and new ideas for information discovery. Therefore, the flexibility to make layout changes to relational databases that represent repository catalogs is likely to be of special interest to librarians and archivists. For the foreseeable future, accomplishing this will require high expertise,[33] necessitating custodian/database administrator collaboration.

Trustworthy Digital Object (TDO) Architecture

For a record to be suitable for preservation, it must contain provenance information or be linked to such information, and the whole record must be protected against undetected tampering. We intuitively want to convey something along the lines of, "At such and such a time, John Doe communicated a document X to the distribution list Y." A TDO thesis is that authoritative metadata should be bound tightly to each preserved object.[34] Derivatives can be

---

[32] Don Chamberlin, *Using the New DB2: IBM's Object-Relational Database System* (San Francisco: Morgan Kaufman, 1996).

[33] Database administrator courses are described in the *Trade Schools, Colleges, and Universities Database Administration School Directory* at http://www.trade-schools.net/directory/database-administrator.asp, accessed 22 April 2009.

[34] A TDO corresponds to the InterPARES notion of a digital component, viz., "a digital object that contains all or part of the content of an electronic record, and/or data or metadata necessary to order, structure, or manifest the content, and that requires specific methods for preservation," quoted in Reagan Moore, "Building Preservation Environments with Data Grid Technology," *American Archivist* 69, no. 1 (2007): 139–58. As for the payloads of preserved objects, the choice of metadata is information creators' prerogative and responsibility. Choices have for many years been discussed within professional societies without such discussions yet having yielded accepted conventions. Almost surely, these conventions will be different for different information genera and

extracted from full records to create repository catalogs with only modest software additions to repository implementations. The idea is to package source information collections so that

1. The bit-string set that represents a work is XML-packaged with registered schema.
2. Each bit-string that represents part of the work is encoded in a computing-platform-independent representation or is accompanied by a bit-string encoded for everlasting intelligibility.
3. Integrity is assured by cryptographic message authentication.[35]
4. The package includes provenance evidence, technical metadata, and one or more identifiers of the object itself.
5. Links to contextual information are secured by cryptographic message authentication codes of the linked entities.
6. Information loss is minimized by replication in mutually independent repositories.
7. Cryptographic signatures are grounded in keys that widely trusted institutions publish periodically.

Conflicting updates are a notorious problem because they add to computer users' workloads. To the extent possible without hampering individual initiative, it is helpful to designate a single preferred location for each information update. The obvious place for a change to information about any object is in some TDO version of that object. Derivative information, such as that in library and archival catalogs, can be created and synchronized, often if necessary, by machine processes.

The TDO structure offers options for evidence of authenticity, contextual information, and whatever might make an object self-describing. Its objective is to enable all reasonable creators' and custodians' choices, rather than to prescribe what choices information creators should make. From an OAIS perspective, a TDO is a Submission Information Package (SIP, as suggested in Figure 1) and also an Archival Information Package (AIP).[36]

TDO Structure

> The authenticity of *electronic records* must be verifiable from elements of the records (i.e., either on their face or linked to them) and contextual to the records (i.e., belonging to their documentary, administrative or technological context), while the authenticity of *electronic copies* of authentic electronic records is attested by the preserver, who has taken responsibility for the process of reproduction....In other words, *any electronic copy of an authentic electronic record is authentic if declared to be so by an officer entrusted with such function*, namely the official preserver.[37]

---

different social situations. The issues are outside the technical structure discussed in the current article, except that it must accommodate every possible choice. Doing so motivated the structure of the Figure 4 relationship block, the TDO ability to include arbitrary metadata blocks, and its internal linking support.

35 Donald Eastlake and Kitty Niles, *Secure XML: The New Syntax for Signatures and Encryption* (Boston: Addison Wesley, 2002); Filip Boudrez, "Digital Signatures and Electronic Records," *Archival Science* 7, no. 2 (2007): 179–93; Anna Lysyanskaya, "How to Keep Secrets Safe," *Scientific American* 299, no. 3 (2008): 89–95.

36 CCDS 2001, loc. cit. §1.7.

37 Heather MacNeil, "Providing Grounds for Trust: Developing Conceptual Requirements for the Long-Term Preservation of Authentic Electronic Records," *Archivaria* 50 (2000): 52–78. The quotation is from page 68, and its italics occur in the original.

Any preservation action begins by collecting and organizing the information to be preserved. As usual, whenever one is planning to replace human steps by machine procedures, one must consider every step explicitly, handling its syntax automatically, thereby allowing human archivists to focus on subjective aspects.

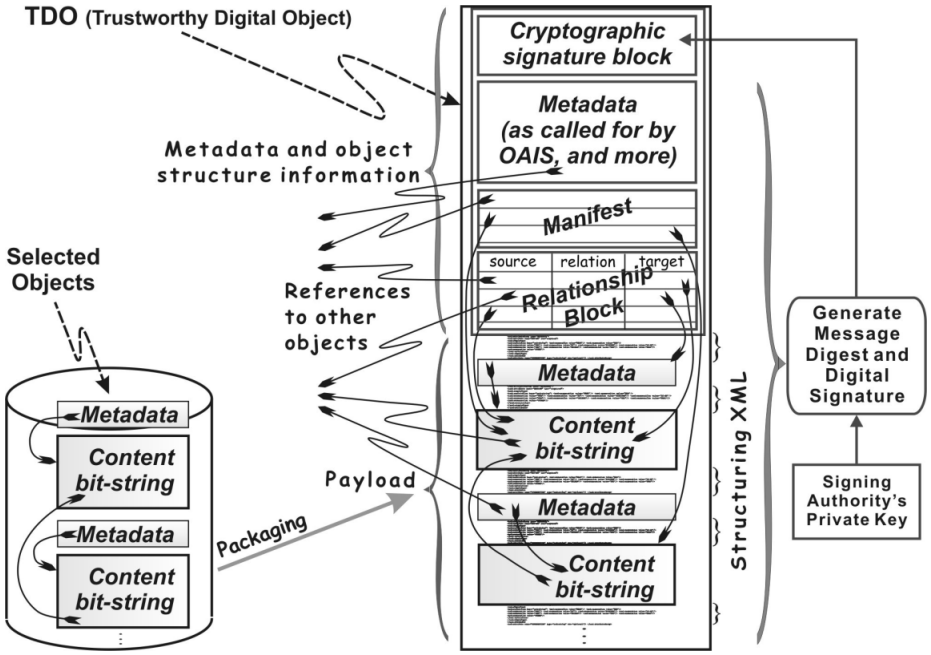

Figure 4: Data to be preserved and a corresponding TDO. Arrows represent contextual links. Each content bit-string might be a single archival record or, alternatively, a sequence of such archival records. Such bit-strings should be faithful copies of archival contents held before the TDO was created.

A creator might choose to preserve a scholarly manuscript, an artistic performance, an engineering specification, a medical patient history, a purchase order for goods or services, a computer program together with its documentation, a description of a set of business transactions—or the description of any idea or event that somebody wants to save as a part of human history. Figure 4 suggests steps in preparing information to be saved for a long time, including steps needed to provide authenticity evidence. Each of its major steps has a close analogy in the world of paper records.[38]

As suggested by the left portion of Figure 4, what is to be preserved is generally a collection of records that some creator or records keeper decides are sufficiently closely related to be packaged as a single entity. The TDO scheme makes no structural distinction between a work of independent authorship and a collection of records. Records will usually not be interpretable without contextual information stored in other records. The archivist must choose among bundling each such contextual entity as part of what will be preserved, bundling a reference (a Figure 4 link) to the entity, or ignoring this piece of context.[39]

Whether the archivist includes a contextual entity chosen directly or as a reference will depend on many factors, including its pertinence domain. For instance, if the root object is a map

---

[38] Compare Filip Boudrez, *Digital Containers for Shipment Into the Future*, in *Electronic Records Supporting e-Government and Digital Archives* (DLM Forum, 2005), available online at http://www.expertisecentrumdavid.be/docs/digital_containers.pdf on 13 June 2009..

[39] Ignoring context might seem irresponsible. However, since contextual dependencies are recursive, an attempt at complete context might call for infinite information. How much context is enough will be a subjective decision.

of Rome, it might include a table of street coordinates directly, but is likely to include needed image-rendering software only by reference because that software is context for many other city maps.

Each person that adds to or updates a work being prepared for preservation can nest or link the initial version, thereby creating a reliable history.

Metadata and Object Self-Description

The cryptographic signature block in Figure 4 describes essential and optional information related to signing, perhaps extending, what is specified for the X.509 standard.[40] The essential information is a time stamp, a signature algorithm identifier, a signing authority identifier, and the signing authority's public key value. Each optional item can be any scalar value, including the signing authority's name, address, email address at the time of signing, a date beyond which the signer believes the content will not be useful, and a text specification of what TDO properties are certified and what facts and commitments are not certified, such as liability disclaimers.

Protection information might be extended by documents that contribute integrity and provenance evidence, such as information about digital watermarks and fingerprints applied to payload elements. As with a ship's cargo, the *payload* of a digital document is the content whose inclusion is the reason for conveying the document from its creator to its recipient.

The manifest and relationship blocks in Figure 4 allow the TDO creator to describe anything about its structure, without disturbing any content bit-string. Each manifest element is a labeled value set describing the corresponding payload block. A *labeled value set* is a structure of the kind called a record in the Cobol programming language, e.g., "name: John Doe, sex: male, citizenship: Canadian, city: Toronto,..." The *n*th manifest element describes the *n*th payload block, starting with the bit offset of this block from the beginning of the TDO, and including whatever additional information its writer chooses. In this article, the word *writer* denotes a human role or surrogate depicted in Figure 1, being some entity that creates or edits information for others. The word is used to suggest that, in the discussion at hand, details of information genre are unimportant. A writer might be a machine process controlled by rules.

The relationship block in Figure 4 is a triadic relation (a table of three-cell rows).[41] The first and last cells each identify some bit-string in the TDO or some external object, or each is a bookmark into such an object. Each value could be an object locator such as a Web address, but will be more durably useful if it is an identifier or a bookmark as described in the next section. The middle cell, describing the relationship between these objects, is either a scalar value or encoded as a labeled value set that might identify further objects and might also include descriptions of the first and last cells.

Every Figure 4 TDO block is optional, including metadata blocks. Much could be written about metadata; indeed, so much has been written that little needs to be added here. Readers can learn more about the specification of the kinds of metadata to be included from the OAIS

---

[40] Ed Gerck, *Overview of Certification Systems: X.509, PKIX, CA, PGP, SKIP* (2000), available at www.thebell.net/papers/certover.pdf, accessed 13 June 2009.

[41] A triadic relation is a concise and convenient way of representing any structure whatsoever. Its fundamental role is described by Rudolf Carnap, *Logical Structure of the World* (PLACE OF PUB AND PUBLISHER, 1928). It is convenient because, in database form, it can be manipulated with SQL statements. It has recently been described in the *Resource Description Framework* standard, available at http://www.w3.org/RDF/, accessed 22 April 2009.

specification,[42] from an early textbook,[43] from the PREMIS dictionary for preservation metadata,[44] from a 2005 review,[45] and from the Library of Congress's *Metadata Encoding and Transmission Standard (METS)*.[46] Any metadata should identify its own schema, because settling on a single worldwide scheme seems unlikely in the foreseeable future.[47]

Object Identifiers and Bookmarks

A *name* is a character string used to allude to an object or set of objects, but might be ambiguous within its usage context, as is "John Smith" in a Chicago telephone directory. An *identifier* is a special kind of name—one without ambiguity.[48] Within its context, it refers either to one object exactly or to no object whatsoever. A unique, universal identifier (UUID) is a special kind of identifier whose context is the union of all contexts, that is, the set of all entities that might be identified. A UUID should not be reused even if the object that it denotes disappears. If a bibliographic citation is unambiguous, it can be used as an identifier. For instance, the string "Bertrand Russell, 'On Denoting'," *Mind* 14 (1905): 479–93", is an identifier. And in 2008, the string "http://cscs.umich.edu/~crshalizi/Russell/denoting/" was a good locator for a copy of this classic essay.

For an identifier to be useful, its context must include a means of choosing new identifiers that do not reproduce previously used identifiers and a resolver (also known as a resolution mechanism)—a means by which any potential user can find a copy of the entity it alludes to. The information base of such a resolver is a special kind of catalog that maps each identifier to the location of what it identifies and perhaps also to object characteristics such as its size, type, access controls, and so on.

Many digital objects will have shared prior versions. We can signal this kind of relationship by a shared attribute that has the semantics of an identifier—a *digital resource identifier* (DRI). An identifier can denote things other than single objects, as well as things that change over time. For instance, the identifier "the British fleet" unambiguously denotes a set of objects whose membership changes with time and whose members are widely distributed, moving about more or less continuously.

Furthermore, it will often be helpful to define identifiers for objects that do not yet exist. And any object can have more than one identifier. In particular, we recommend that each TDO include its own UUID and also a DRI. As will be apparent from the next section, pervasive use of DRIs would help readers discover the history of information that interests them.

How can one avoid a cumbersome system of ensuring that the next identifier choice does not reproduce some previous choice? The simplest way is to use a random number generator to

---

42 CCSDS, *Reference Model for an Open Archival Information System*, §4.16.

43 Susan S. Lazinger, *Digital Preservation and Metadata: History, Theory, Practice* (PLACE OF PUBLICATION, Greenwood, 2001).

44 Recent PREMIS activity is described at http://www.oclc.org/research/projects/pmwg/, accessed 22 April 2009.

45 Brian Lavoie and Richard Gartner, *Preservation Metadata*, DPC Technology Watch Report 05-01 (2005), http://www.dpconline.org/docs/reports/dpctw05-01.pdf, accessed 22 April 2009.

46 Judith Pearce, David Pearson, Megan Williams, and Scott Yeadon, "The Australian *METS* Profile: A Journey about Metadata," *D-Lib Magazine* 14 (March/April 2008).

47 Lois Mai Chan and Marcia Lei Zeng, "Metadata Interoperability and Standardization—A Study of Methodology" (in two parts), *D-Lib Magazine* 12, no. 6, (2006).

48 Giuseppe Vitiello, "Identifiers and Identification Systems: An Informational Look at Policies and Roles from a Library Perspective," *D-Lib Magazine* 10, no. 1 (2004).

choose each character of a fairly long bit-string. For instance, a string of 32 ASCII characters randomly chosen from the lowercase Roman alphabet and numerals has probability less than 10^{-39} of colliding with any of a billion similarly chosen identifiers, and no chance of colliding with any shorter or longer identifier.

In principle, nothing more is required, not even easy human readability. In practice, limitations of widely used computing protocols and pre-existing identifier systems are best accommodated by a few simple syntactic constraints.[49] For instance, many identifier schemes start with a prefix that disambiguates identifiers that use different resolvers, such as "ISBN:," "uri:," and "http:."[50] To accommodate legacy software, the characters used in an identifier can be limited to lowercase ASCII alphanumerics ([a–z 0–9]) and a few punctuation marks ([: / #]). For easy readability, printed identifier representations often include blanks or hyphens that are not part of their internal representations; ISBN numbers, for example.

Any identifier can be made into a bookmark by appending a "#" followed by an integer denoting a bit offset into the object identified. Such a bookmark can be made to indicate an extent by appending a second "#" and integer. Alternatively, for data types such as text documents with standard formats that include a syntax for bookmarks, their bookmark strings can be used instead of integers.

Object Versions and Audit Trails

From an archivist's perspective, the significant events in a record's history are its transfers from a creator to each successor in a chain of custody (handoff events). These are the only occasions when two people assuredly have access to the same fixed version of the work. TDOs can be nested. To create an audit trail, each successor could include in a TDO he or she creates a copy of the TDO as received. If every successor does this, the latest TDO will reflect the entire history of the work.[51] Alternatively, a successor might embed a reliable link to the prior version in each new TDO instead of a prior version copy.

By whom and how a TDO should be constructed is not prescribed as part of TDO methodology. Instead, we are designing a tool intended to make TDO construction easy for almost anybody. TDO structure allows provenance information, authenticity certification, and object relationships to be tailored to meet different participants' subjective notions of what is good enough. For instance, in some cases, custodians will want to include evidence of a handoff event.

Each participant in a TDO creation sequence usually is, or readily can become, acquainted with his or her predecessor and successor. Thus the public keys that validate authorized version deliveries can readily be shared without depending on a Public Key Infrastructure (PKI) certificate authority. This arrangement avoids well-known PKI security risks, such as the disappearance of certificate authorities.

How Archival Requirements Are Met

The special concerns of archivists can be handled by appropriate metadata and by judicious choices of references suggested by the arrows in Figure 3. Most archival

---

[49] For a careful discussion, see Gladney, *Preserving Digital Information*, §7.3.

[50] Some identifiers are used mostly in contexts that suggest their nature. An example is Social Security numbers.

[51] Compare what is written about annotations by MacNeil, "Providing Grounds for Trust",,64.

requirements[52] are easily handled by well-known technology as suggested in Table 2. A critical assumption—that it will always be possible to copy bit-strings from some soon-to-be-obsolete machine environment to some environment that will be reliable for several years longer than the expiring machinery—seems quite safe. There are only two conceptual challenges. Each can be addressed with technology not available for information on paper or impractical for some kinds of records, particularly for huge numbers of records.

Table 2. Meeting Archival Requirements

Retrieve saved information.	Addressed by many near-term content management offerings and therefore not treated in this review.
Render or use content as its writers intended.	See below.
Judge information trustworthiness.	See below.
Exploit embedded links reliably.	See below.
Avoid hindering users by technical complexity.	This issue cannot satisfactorily be discussed, but must be shown in implementations for users' critical approval.
Autonomy in packaging content for preservation.	See below.
Ingest content into archives.	Requires additions to the Figure 1 digital record import module and addressed in metadata sharing protocols.[53]
Continue using currently deployed content management software.	Achieved by having most software enhancements for LDP inherent in TDO editors and viewing programs instead of enhancements to repository software.
Share information without adjustments requiring human judgment.	Achieved by using widely accepted conventions for TDO structure and metadata.
Share LDP effort by all stakeholders.	Achieved by making TDO editing easy for ordinary users as well as for repository personnel.
Ensure that saved information survives any repository's demise.	Achievable by each writer submitting his TDOs to several autonomous archives.

The first challenge is how to ensure that eventual users can understand and use preserved information as well as can current users, even though computing technology will have changed and eventual users cannot seek clarification of obscure points by asking today's creators and custodians. A few relatively simple and widely used information formats can be handled by EDP standards. All the rest can be handled with the assistance of computing machine emulators. The

---

[52] See, for example, Heather MacNeil, "Providing Grounds for Trust"; Maria Guercio, "Principles, Methods and Instruments for the Creation, Preservation and Use of Archival Records in the Digital Environment," *American Archivist* 64 (2001): 238–69. Anne J. Gilliland-Swetland, *The Archival Paradigm—the Genesis and Rationales and Evolution of Archival Principles and Practices*, in *Enduring Paradigm, New Opportunities: The Value of the Archival Perspective in the Digital Environment*, CLIR Report 89 (2000), http://www.clir.org/pubs/reports/pub89/archival.html, accessed 22 April 2009. Also Tibbo, "On the Nature and Importance of Archiving," §4.

[53] For instance, see *The Open Archives Initiative Protocol for Metadata Harvesting, 2004,* described at http://www.openarchives.org/OAI/openarchivesprotocol.html.

second challenge is how to provide reliable evidence for the integrity and authenticity of preserved information, particularly for records that are tempting targets for malevolent modification. This yields to modern asymmetric cryptography combined with infrastructure for handling secret keys.

The principal programming action needed is creation of an editor to create TDOs, to inspect TDOs, and to extract TDO contents into forms acceptable to existing information processing programs. Such an editor will be a specialized kind of XML editor.

Encoding Information To Be Durably Intelligible

> [T]oo much attention has been devoted to ensuring access to electronic records fifty or one hundred years from now when we have no way of forecasting what kinds of technology will be available then. Instead, we should focus on a much shorter time frame, perhaps on the order of ten to twenty years or so, during which time information technologies are likely to be relatively stable.[54]

Unlike a physical record, users cannot extract digital information without a computer and, seeing it, would not know its meaning, since most cannot make sense of bit-strings. However, following Dollar's recommendation would be neither prudent or necessary. It would be imprudent because there are no indications that software is stabilizing. In fact, after half of Dollar's buffer period has passed, software innovation seems faster than ever. Dollar's recommendation is unnecessary because correct information rendering is possible without technology forecasting and without format migration that has been proposed, but criticized as unreliable.[55]

For simple files, standards independent of ephemeral technology suffice, since these are specified precisely and intelligibly. Word processor files have recently become available in XML format, but their schema specifications have not yet been sufficiently carefully considered to be deemed prudent for LDP.

JPEG[56] and PDF[57] are two widely used formats for which using standards is proposed. The case of PDF illustrates how difficult it is to work with complex formats and how long it takes for a format standard to become accepted. For files presumed durable by way of a format standard, there is an implicit assumption that programs for this file type will forever have functionality at least equivalent to today's.

For more complex digital objects, we create emulator programs that accompany today's content to render it for our descendants. This uses current hardware and software to create rewrite routines whose outputs include all the essential information from saved bit-strings. We write these transformation programs in the instruction set of a relatively simple virtual machine.

---

[54] Dollar, *Authentic Electronic Records*, Introduction p.5.

[55] Tibbo, "On the Nature and Importance of Archiving," §4.3.

[56] Paolo Buonora and Franco Liberati, "A Format for Digital Preservation of Images: A Study on JPEG 2000 File Robustness," *D-Lib Magazine* 14 (July 2008).

[57] PDF is, in fact, a complicated topic treated by several standards. See descriptions of ISO 19005 (PDF/A, 2005) and ISO 32000-1 (PDF 1.7, 2008). See also Sabine Schrimpf, "Standardization in the Area of Digital Long-Term Preservation," presentation at Archiving 2008 Conference, PLACE.

Theory of Virtual Machines

The underlying theory is the Church-Turing thesis, a subtly difficult idea informally expressed as "a certain very simple kind of machine can accomplish any feasible computation described with a finite set of rules."[58] Alan Turing devised the prototypical machine when still a Cambridge undergraduate.[59] We can rephrase Church-Turing as "any transformation from an inconvenient record format to a convenient format can be accomplished by a Turing machine."

It is easy to program any practical computer to emulate a Turing machine. If we write an interesting transformation as a Turing machine program, we can today test that this program is correct by executing it on some current computer and also can be confident that it can be correctly executed on any future computer. The feasibility of such correctness testing is the key to our not needing to know how future computers will be different from today's machines.

Writing interesting transformations as Turing machine programs would be tedious because the Turing machine manipulates only a single bit at each step. It is, however, possible to devise a more practical machine that is as powerful as a Turing machine. Raymond Lorie has designed such a "Turing equivalent" machine that he calls a Universal Virtual Computer (UVC).[60] The UVC definition is simple enough for its complete specification to take only eleven short pages.[61] This specification will surely be correctly understood whenever needed. This inspires confidence that the UVC can be correctly emulated by any future real computer.

Using Emulator Encoding

After a skilled programmer has created a tool set for that bit-string's format, a necessary step for each file format to be preserved, it is simple for end users to prepare a bit-string for reliable future intelligibility. How such a tool set will handle a file that is part of some work to be preserved is suggested by Figure 5, which depicts transforming data for preservation, copying the bit-string that represents a TDO from time to time to forestall the effect of hardware obsolescence, and preparing the preserved information for use a century from now. In the figure and text, "2009" is used as an abbreviation for "at the present time" and "2109" for "many decades from now."

---

58 *Stanford Encyclopedia of Philosophy*, s.v., "Church-Turing Thesis," http://plato.stanford.edu/entries/church-turing/, accessed 22 April 2009.

59 Andrew Hodges, *Alan Turing: The Enigma* (Vintage: London, 1992), chapter 2.

60 Raymond A. Lorie and Raymond J. van Diessen, *A Universal Virtual Computer for Long-term Preservation of Digital Information*, IBM Research Report RJ 10338 (February 2005). A proof-of-concept UVC implementation is available from IBM at http://www.alphaworks.ibm.com/tech/uvc, accessed 22 April 2009.

61 Gladney, *Preserving Digital Information*, Appendix E: Universal Virtual Computer Specification.

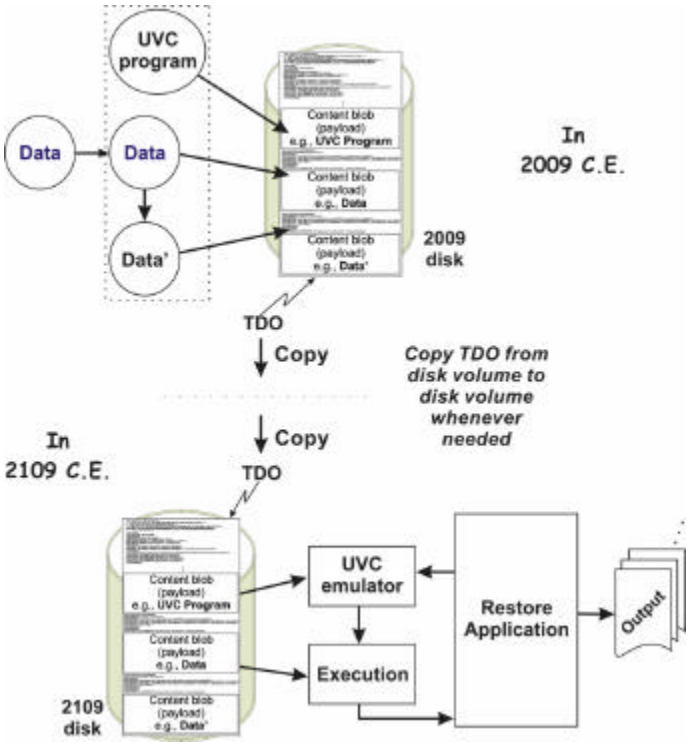

Figure 5 top: Encoding a bit-string and embedding the results into a TDO today;
middle: copying it whenever a storage medium or repository needs to be replaced;
bottom: extracting from the TDO and rendering essential content a century from today.

The idea is to save into a TDO the original data in whatever form it was produced or was used in 2009, together with a UVC program that creates durably intelligible or otherwise useful renderings. Part of a TDO editor is an appendage for each supported input file type. As suggested by the figure, this appendage accepts a file Data and an indicator of its file type, finds the UVC program for that file type, and embeds this and Data into the TDO being produced. Optionally, to simplify the programming work, the editor transforms the input Data to one or more Data' instances that are also embedded into the TDO. For instance, we might use OpenOffice software to convert a Microsoft Word document into its Open Document Format (ODF) equivalent because no restrictive copyright encumbers the latter. The UVC program, executed whenever an eventual reader seeks access to the information originating in Data (see the bottom third of Figure 5), eliminates dependency on today's software and is designed to show what is essential, signaling accidental aspects as much as possible.[62]

Between 2009 and 2109, the TDO will need to be copied to new storage volumes or repositories whenever the current storage environment approaches obsolescence. Such copying, suggested by the middle portion of Figure 5, should reproduce the TDO bit-string exactly.

A 2109 Restore Application uses a UVC emulator to execute the UVC program. This program parses **Data**, **Data'**, ... and produces some number bit-strings for the Restore Application to use, perhaps to print them as Output files as suggested in lower part of Figure 5. More than one output might be wanted to reduce ambiguity,[62] or to represent information not included in the first strings chosen. The execution might also produce redundant result strings for readers' convenience. For instance, for scientific data, the process might produce a form for

---

[62] See Essential and Accidental Information, §4.1 of Gladney, *Preserving Digital Information.*

printing tables, a file of commands to load a relational database, and instructions for drawing a directed graph.

Programming Required

What do support programmers need to do?

One UVC emulator will be needed for every machine CPU type that will figure in preservation. That is, the world will need only one emulator today for each distinct Intel CPU design, one for IBM S/370, and so on. In future years, the world will need only one emulator for each CPU type used to exploit preserved objects. For each file format to be preserved with UVC assistance, some 2009 programmer will have to write a UVC program and a plug-in appendage to the TDO editor mentioned above. This appendage is to start by creating zero or more Data' bit-strings from an input Data bit-string. For instance, to preserve Microsoft PowerPoint presentations, it might start by converting a PowerPoint bit-string to an OpenOffice Impress (.odp) bit-string (an easy example, because OpenOffice can itself be used). Finally, the appendage needs to insert Data and Data' into the TDO being constructed, together with a copy of the UVC program or a reliable reference (see the next section) to an already existing TDO that preserves this UVC program.

The new UVC program is to rewrite (transform) the information in Data or Data' into one or more bit-strings that will be intelligible to eventual users. Writing the UVC program for a data type is similar to writing any other language transformation program, except that it needs to be written as UVC code. This transformation, or set of transformations, must be written with a view to mitigating ambiguities inherent in essential/accidental information distinctions already described.[63] To accomplish this, the programmer needs to exploit the formal specification for the file format being handled, that is, to have the same sort of knowledge and skills as the many programmers who already created tools for the file type in question.

How difficult and expensive might this task be? Although this is hard to estimate (estimating programming project costs is notoriously difficult), we can say that it is likely to be easier and less costly than creating an interactive editor for the file type. First, the transformation program can be written with the assumption that it will be used only for bit-strings without significant format errors (nobody should preserve flawed information). Therefore the program will not need as strong an error detection and recovery code as other programs for the file kinds at hand, making programming less costly. Second, the program will not need a graphical user interface for file editing.

The programmer can test the correctness of the UVC program by comparing the results of an emulation on a 2009 computer with those from a UVC emulator running on a computer incompatible with the machine on which everything originated, such as an Apple Macintosh for an IBM PC. Another good test would be to translate exemplary results and observe whether independent human users understand the information to be conveyed in the future and find it convenient.

To prepare for using UVC-preserved bit-strings, our 2109 successors must write a UVC emulator that executes on then-current computers. They must also create a Restore application. The latter must invoke the UVC emulator, passing the locations of the saved UVC program, the saved Data and Data' strings (there might be several of the latter), and addresses where results should be stored. It also needs to print or otherwise handle the results.

---

63 Gladney, *Preserving Digital Information,*,§4.1.

Packaging Information To Be Trustworthy

Ideally, any reader could quickly and easily decide whether a preserved record is sufficiently trustworthy for his or her use, doing so without needing human help, even for records that are tempting targets for fraudulent modification. However, some records are susceptible to tiny changes that would evade discovery and that make them misleading. A trivial example illustrates the risk:

> Queen of the Fairies: The law is clear—every fairy must die who marries a mortal!
> Lord Chancellor: Allow me, as an old Equity draftsman, to make a suggestion. The subtleties of the legal mind are equal to the emergency. The thing is really quite simple—the insertion of a single word will do it. Let it stand that every fairy shall die who doesn't marry a mortal, and there you are, out of your difficulty at once![64]

In ancient times, the authenticity of documents was evidenced by wax seals affixed with signet rings. A digital counterpart is a message authentication code firmly bound to a record. Such a cryptographic certificate can itself be authenticated by a recursive certificate chain, using what is called a Web-of-Trust model.[65] In this, each certified signature is itself certified. This recursion is grounded in the published cryptographic key of a widely trusted institution—trusted partly because the institution has little motivation to mislead and would lose business if it lost clients' confidence about its ethics.

The appropriate tool is an asymmetric cryptographic algorithm with a two-part key—a secret part used to encode a signature (or any other kind of message), but not revealed by its user to anybody else, and a public part with which messages can be decoded but not encoded, and which is widely revealed.[66] For instance, an institution certifying large numbers of signatures (directly or indirectly) might publish its public key in news magazines and trade journals. If it does so and changes its key pair annually or more frequently, destroying all records of the private key value, then each published public key becomes evidence of the time period in which the signed keys were certified.

Faithful interpretation of a record can be compromised by unreliable contextual references—other object references suggested by Figure 4. This risk would be forestalled if each reference included the message authentication code of the record that it alludes to. Using this and the cryptographic signature within the contextual record can mitigate the risk.

Standards and Infrastructure Dependencies

A panoply of EDP standards—file format standards, software interface standards, communication protocols—in part enables rapid computing improvements and developer autonomy. Some of these are very complicated. Some have useful lifetimes of only a decade or two before they yield to replacements. Prediction of the specifics even a few years into the future would be difficult. Many such EDP standards are used in content management (see Figure 1). As illustrated for metadata, some such standards and conventions are still under discussion and

---

64 W. S. Gilbert and A. Sullivan, *Iolanthe,* act 2, 1882.

65 Germano Caronni, "Walking the Web of Trust," *Proceedings of the 9th Workshop on Enabling Technologies,* IEEE Computer Society Press (2000), available at http://www.olymp.org/~caronni/work/papers/wetice-web-final.pdf, accessed 22 April 2009.

66 Eastlake and Niles, *Secure XML,* §2.4.

development.[67] The march toward consensus about best choices is a social process that neither can nor should be hurried.

One can, however, already choose information packaging that will be reliably useful a century from now. Refinements will make better choices possible, so that future readers will find it easier to use information packaged twenty years from now than information packaged today. For the time being, however, file format specifications are generally too weak, or at least too little understood, to be relied on to ensure that the files they describe will be correctly interpreted half a century from now.[68]

In contrast to aggressive use of standards in near-term content management services, we believe that LDP software should be based on a relatively small number of basic and mature standards. In addition to file format standards that are thought to be sufficiently simple and stable, these should include Unicode/UCS[69] and UTF-8 for character encoding,[70] a small subset of XML standards,[71] and a few others, such as *Abstract Syntax Notation One* (ASN.1)[72] for describing other standards. Although it is premature to identify a complete list, we know enough to proceed confidently, perhaps discovering a few additions as we implement TDO editing.

Digital archiving service depends on a great deal of infrastructure. This includes tools for search index and metadata extraction from digital documents, search infrastructure and collection catalog creation, management of cryptographic keys and certificates, format registries, data display and editing programs, and many other components.[73] Our LDP methodology assumes such tools will always exist, but does not depend on specific methodological details that might change.

Discussion

> At this time there is no clear solution to the challenges of technological obsolescence. Both...approaches and those that fall between them in Thibodeau's classification…require and deserve extensive research and development. It is essential that the archival perspective that stresses preservation of authenticity and reliability of records and information be present in all such research if long-term digital archiving is to become a reality.[74]

---

[67] Cloonan and Sanett, "Preservation Strategies," §J, mentions the Generalized International Standard Archival Description, Encoded Archival Description, MARC, modified Library of Congress Subject Headings, and the Dublin Core as exemplary metadata standards. See also Rebecca S. Guenther, "Battle of the Buzzwords: Flexibility vs. Interoperability when Implementing PREMIS in METS," *D-Lib Magazine* 14 (July 2008).

[68] This opinion depends on E. P. McLellan, *Selecting Digital File Formats for Preservation*, InterPARES 2 Project (2007), available at http://www.interpares.org/display_file.cfm?doc=ip2_file_formats(complete).pdf, accessed 22 April 2009.

[69] The Unicode Standard is described at http://www.unicode.org/standard/standard.html, accessed 22 April 2009.

[70] For UTF-8, see www.unicode.org/standard/standard.html /, accessed 22 April 2009.

[71] XML 1.0, XML Namespaces, XPath, and XPointer are the core needed by information consumers' agents. See IBM 2004, *A Survey of XML Standards*, available at http://www-128.ibm.com/developerworks/xml/library/x-stand1.html, accessed 22 April 2009.

[72] B.S. Kaliski, Jr., *A Layman's Guide to a Subset of ASN.1, BER, and DER* (1993), available at http://luca.ntop.org/Teaching/Appunti/asn1.html, accessed 22 April 2009.

[73] Those listed are suggested by recent conferences such as the 2008 Joint Conference on Digital Libraries, whose program is available at http://www.jcdl2008.org/sessions.html, accessed 22 April 2009.

[74] Tibbo 2003 loc. cit. §6.4. The allusion is to Kenneth Thibodeau, 2002, loc. cit.

TDO methodology provides the solution Tibbo calls for. Our paper might be viewed as very technical by some archivists. However, if archivists want to participate in and influence how future archives are managed, they need to understand technical structure (but not most technical details). TDO structure and representation are intended to capture those aspects susceptible to automation and to permit all prescribed practice for the semantic aspects that depend on human judgment.

We believe that the methodology sketched in this review is entirely consistent with archival principles described by Dollar,[2] Tibbo,[14] Gilliland,[52] and Duranti.[75] The key LDP requirements are expressed above in terms of what end users, both archivists and clients of archiving services, will want. The additional engineering requirements identified above—for matters such as scaling, service reliability, integration with related software, and so on—prove to be much the same as those of any digital library application. Since these have been solved in principle some years ago, they have not been treated again. Of course, performance optimization and reliability engineering continue, but these aspects are of less interest to archivists than issues of long-term preservation.

The amount of new computer code needed for LDP is small compared to the amount of information to be preserved, and also small compared to other document management software. The central component is a TDO editor that is not much more than a specialized XML editor. A single UVC definition can be sufficient for all data types and for all time. A single set of UVC programs can be sufficient for each type of file that people might want to preserve. A single UVC emulator is all that is needed for each computer architecture of interest. Everything else needed can be accomplished by packaging code already available; much of that is open-source software. In short, long-term digital preservation can be managed as a modest extension of near-term services offered by many commercial and public sector CM offerings (also known as digital repository services).

What about Trusted Digital Repositories?

Most digital preservation work has been directed toward Trusted Digital Repositories (TDR) methodology, a widely discussed approach to digital preservation. A careful comparison of TDR and TDO methodology is being prepared in a separate article.

> The third fundamental issue associated with the authenticity of electronic records is the assurance that a trusted third party is responsible for...ensuring that [stored electronic records] remain unaltered. This means that the creators/users of electronic records will transfer them to the custody of a trusted third party where they can be used under existing access rules and regulations but cannot be changed by anyone,...This "trusted third party" can be a records repository, an organization's archives, a public archive, or a service bureau that adheres to "best archival practices"...[76]

Trust between individuals and institutions is a complex topic beginning to receive direct attention.[77] Although trust might not be a critical concern for scholarly and cultural works, it

---

[75] Luciana Duranti, *The Long-Term Preservation of the Dynamic and Interactive Records of the Arts, Sciences and E-Government: InterPARES 2*, *Documents Numérique* 8, no. 1, 2004: 1–14; Duranti, *Long-term Preservation of Authentic Electronic Records: Findings of the InterPARES Project* (PLACE OF PUBLICATION, Archilab, 2005).

[76] Dollar, *Authentic Electronic Records*, §1.3.3.

[77] F. Berman, A. Kozbial, R. H. McDonald, and B. Schottlaender, "The Need for Formalized Trust in Digital Repository Collaborative Infrastructure," *Proceedings of the NSF/JISC Repository Workshop* (2007).

certainly is critical for legal documents, financial records, and personal privacy. In view of massive Internet chicanery, TDRs as specified today[78] do not seem to be prudent repositories for sensitive information. Furthermore, for future readers to decide whether or not to trust information received from a TDR network, they need to know a great deal about that network and be skilled at evaluating that knowledge. It is unlikely that most readers will have the necessary patience or expertise.

The TDO approach shifts the locus of trust to a relatively small set of public cryptographic keys and to the adequacy of the metadata bound to preserved objects. Given readily provided tools and instructions, any reader will be able to judge whether information has been distorted, doing so quickly without any immediate human help. Thus, asymmetric cryptography is a better prospect for authenticity management of publicly accessible digital records than any alternative described in archiving literature, with the possible exception of the internal procedures of a few well-managed central government archives. Although it is not thought foolproof, it is much better understood than trust in institutions and continues to be an active research topic.

Next Steps and Future Developments

We believe TDO methodology to be correct and, in principle, sufficient for preserving anything that can be preserved digitally, including so-called dynamic information,[79] although TDO tools that archivists find practical and convenient still need to be written.[80]

Intellectual property rights pose a bigger barrier than technical challenges.[81] Source code for lucrative products, such as Microsoft Office, is typically held as a trade secret and is always circumscribed by copyright protection. Even if source code were available, Microsoft would almost surely oppose translation for preservation as a violation of its copyright privilege of creating a derivative work. As long as business conditions remain as they are today, upgrade versions of Microsoft Office will be a lucrative revenue source. In fact, Microsoft might justifiably argue that such future versions will provide what is needed for preservation of Microsoft Office files.

The next step is a pilot implementation. Readers might reasonably ask why this was not done some time ago. One of two reasons is that, until recently, we did not understand the content management context and LDP design sufficiently to be confident that the TDO design was correct and sufficient for archival records. One school of thought holds that roughly 80 percent of any project's time budget should be spent on design before implementation begins. The second reason is that resources to build and deploy production-quality software have not been available.

---

78 RLG-NARA Digital Repository Certification Task Force, *Trustworthy Repositories Audit & Certification (TRAC): Criteria and Checklist* (2007), available at http://www.crl.edu/PDF/trac.pdf, accessed 22 April 2009.

79 For a discussion of philosophical issues around the word *dynamic*, see Gladney, *Preserving Digital Information*, §5.4.

80 The Koninklijke Bibliotheek and IBM have built a prototype for part of what is needed. See J. R. van der Hoeven, R .J. van Diessen, and K. van der Meer, "Development of a Universal Virtual Computer (UVC) for Long-Term Preservation of Digital Objects," *Journal of Information Science* 31, no. 3 (2005): 196–208.

81 Pamela Samuelson, "Intellectual Property for an Information Age," *Communications of the ACM* 44, no. 2, 67–68 (2001); Catherine Ayre and Adrienne Muir, "The Right to Preserve," *D-Lib Magazine* 10, no. 3 (2004).

Challenges to Skeptics

> *The Preservation of the Integrity of Electronic Records* [project] goal was to identify and define conceptually the nature of an electronic record and the conditions necessary to ensure its integrity, meaning its reliability and authenticity, during its active and semi-active life. The research resulted in...rules for developing and implementing a trustworthy electronic record-keeping system.[82]
>
> The major dilemmas of digital preservation have yet to be satisfactorily solved...and the development of a "magic bullet" universal solution is unlikely to appear in the near future.[83]

Karen Gracy echoes librarians' admonitions not to expect a "silver bullet" solution. If these commentators mean that a solution is an existing software package that contains everything needed for a digital archive for every kind of data, they are correct. No such package exists today, or is ever likely to exist, because different institutions have different needs and preferences, and because there are so many different data formats, with new formats always likely to appear. No software engineer would talk about a "single solution," because the phrase has little sense for computing procedures.

We suggest that the TDO offers a concise prescription for the technical portion of LDP. It will provide a framework and toolkit from which satisfactory and inexpensive archiving support can easily be assembled for any specific situation. Furthermore, it can readily be extended to accommodate new data types and new needs. Needed now is what, in IBM Research, used to be called SMOP—"a simple matter of programming."

Librarians and archivists need to vet this solution or find a better one, they need to find funding to implement whatever is eventually approved, and they need to work with software engineers to ensure that what is built is truly what they want. Taking responsibility in this fashion has been called for:

> [T]he participants issued a series of resolutions, calling for greater involvement by record keepers in information technology initiatives;...collaborative approaches to records and information technology projects;...and the continued development...of standards for electronic records management.[84]

Conclusions

For professional and social reasons, we want to deploy infrastructure for preserving any digital content whatsoever in ways that meet the needs of its eventual users. This article has sketched and reviewed software that would satisfy every requirement identified in archival and library literature. Its core is architecture for rendering information in any data format whatsoever and for protecting at-risk information.

An unmatched strength of TDO structure—if implanted and managed correctly—is that any reader will be able to judge whether a delivered object is trustworthy. Readers need not worry that procedures hidden in the internal workings of archives might be flawed or might not have been faithfully and correctly executed over the decades or centuries since some artifact was

---

[82] MacNeil, "Trusting Records", §4.3.

[83] Karen F. Gracy, review of "Preserving Digital Materials," by Ross Harvey, *American Archivist* 69, no. 2, (2006): 535.

[84] Laura Millar, *Authenticity of Electronic Records*, report for UNESCO and International Council on Archives (2004), available at http://www.ica.org/en/node/30209, accessed 22 April 2009.

created. Everything needed for trustworthiness evaluation is in each TDO or in objects to which it links recursively. By this means it would achieve much-sought information transparency.

TDO methodology will also make information preservation possible for anybody, not only repository custodians. This is desirable partly because creators of future custodial holdings commonly know much better than librarians or archivists why their material is valuable, what they want to communicate and to whom, and what the pertinent origin, history, and relationships to the world are. TDO methodology can also help to make LDP affordable and help people handle the immense number of preservation-worthy records. It will achieve maximum autonomy for every communication participant. Finally, TDO methodology conforms to time-honored archival principles.

Widespread long-term digital preservation will not occur before the cost of preserving information is small compared to the cost of creating it. A good TDO implementation would help accomplish this.