

Towards smart storage for repository preservation services

Steve Hitchcock, David Tarrant, Adrian Brown¹, Ben O'Steen², Neil Jefferies² and Leslie Carr

IAM Group, School of Electronics
and Computer Science, University of
Southampton, SO17 1BJ, UK
{D.Tarrant, S.Hitchcock,
L.Carr}@ecs.soton.ac.uk

¹ The National Archives, Kew,
Richmond, Surrey, TW9 4DU, UK
adrian.brown@nationalarchives.
gov.uk

² Oxford University Library
Services, Systems and Electronic
Resources Service, Osney One
Building, Osney Mead, Oxford OX2
0EW, UK
{Benjamin.Osteen,
neil.jefferies}@sers.ox.ac.uk

Abstract

The move to digital is being accompanied by a huge rise in volumes of (born-digital) content and data. As a result the curation lifecycle has to be redrawn. Processes such as selection and evaluation for preservation have to be driven by automation. Manual processes will not scale, and the traditional signifiers and selection criteria in older formats, such as print publication, are changing. The paper will examine at a conceptual and practical level how preservation intelligence can be built into software-based digital preservation tools and services on the Web and across the network 'cloud' to create 'smart' storage for long-term, continuous data monitoring and management. Some early examples will be presented, focussing on storage management and format risk assessment.

Digital preservation: the big picture

Digital preservation is dealing with a big picture: "A preservation environment manages communication from the past while communicating with the future" (Moore, 2008). In other words, digital preservation might be concerned with any specified digital data for, and at, any specified time. The classic way of dealing with challenges on this scale is to break these down into manageable processes and activities, as digital preservation practitioners have been doing: storage, managing formats, risk assessment, metadata, trust and provenance, all held together and directed by policy.

The advantage digital has over other forms of data is the ability to reconnect, or reintegrate, these components or services, to fulfil the big picture. In this way specified digital content in various locations can be monitored and acted upon by a series of services provided over the Web. Since at the core of any preservation approach is storage, we call this approach 'smart storage' because it combines an underlying passive storage approach with the intelligence provided through the respective services. The key to realising smart storage, as well as building the services, is to enable the services to share information with the digital content sources they may be acting on. This is done through machine-level application programming interfaces (APIs) and protocols, and has become a focus of the work of the JISC-funded Preserv 2 project [Link 1].

Institutional repositories

One of the drivers for the growth of digital content is the Web. The content the project is concerned with is found in digital repositories, specifically in repositories set up by institutions of higher education and research to manage and disseminate their digital intellectual outputs. These institutional repositories (IRs) are a special type of Web site, typically based on some repository software that presents a database of records pointing to the objects deposited. IRs provide varying degrees of moderation on the entry of content, from membership of the institution to some form of light review. Although there are few examples yet of comprehensive policy for these repositories (Hitchcock *et al.* 2007), it is expected the institutions will take a long-term view and that services will be needed to preserve the materials collected by IRs.

The Preserv 2 project is investigating the provision of preservation services for IRs. Rather than viewing itself as a potential service provider, the project is an enabler. It is identifying how machine interfaces can be supported between emerging preservation tools, services, prospective service providers and IRs.

IRs in flux

However, institutional repositories (IRs) are perhaps in a greater state of flux than at any time since their effective inception in 2000 motivated by the emergence of the Open Archives Initiative (OAI). While the number of IRs and the volume of content are growing, there is uncertainty in terms of target content - published papers, theses, research data, teaching materials - policy, rights, even locus of content and responsibility for long-term management.

IRs are developing alongside subject-oriented repositories, some long-established such as the physics Arxiv, while others such as PubMed Central (and its UK counterpart) have been built to fulfil research funder mandates on the deposit and access to research publications. While ostensibly these different types of repository have common aims, to optimise access to the results of research through open access, how they should

align in terms of content deposit policy, sharing and responsibility for long-term management is still an active discussion (American-Scientist-Open-Access-Forum, 2008a).

When planning and costing long-term data management, open access IRs, those targeting deposit of published research papers, in addition need to take account of author agreements with publishers, and of publishers' arrangements for preservation of this content, often in association with national libraries and driven by legal deposit legislation.

Even the infrastructure of IRs is changing. The majority of IRs are built with open source, OAI-compliant software such as DSpace, EPrints and Fedora. The emergence of OAI-ORE (Object Reuse and Exchange, Lagoze and Van de Sompel, 2008) effectively frees the data from being captive in such systems and reemphasises the role of repository software to provide the most effective interfaces for services and activities, such as content deposit, repository management, and dissemination functions such as search, browse and OAI-PMH. The recent emergence of commercial repository services (RSP 2008), from software-specific services to digital library services or more general 'cloud' or network storage services, is likely to further challenge the conventional view of repositories today as a locally-hosted 'box'. It has even been suggested that the 'institutional' role in the IR will resolve to policy, principally to define the target content and mandate its collection for open access, but without specifying the destination of deposits (American-Scientist-Open-Access-Forum, 2008b).

Against this background, where the content and preservation requirements are effectively not yet specified – for IRs we don't know exactly what type of content will be stored, where, and what policy and rights apply to that content and who exercises responsibility for long-term management – it seems appropriate, then, that we consider the big preservation picture and prepare for when the specifics are known and for all eventualities that might prevail at that time.

Towards smart storage

Two characteristics of digital data management, one that applies particularly to digital repositories, are driving approaches towards preservation goals and begin to suggest approaches that we are attempting to identify as smart storage:

- Scale and economics: the volume of digital data continues to grow rapidly, while the relative cost of storage decreases, to the extent that services that act on data must be automated rather than require substantive manual intervention, and will demand massive, and probably selectable, storage (Wood 2008)
- Interoperability: the viability of IRs is predicated on interoperability provided by the

OAI Protocol for Metadata Harvesting (OAI-PMH), to enable the aggregated contents of repositories to be searched and viewed globally rather than just locally. We now seek to exploit interoperability in the wider context of what is more clearly recognised as the operative Web architecture, known as Representational State Transfer, or RESTful, and is the basis of many Web 2.0 applications that expose and share data

Open storage

In terms of content and data, IRs are characterised by openness: the most widely used repository softwares are open source, and the content in IRs is largely open access. From the outset IRs have been 'open archives' having adopted the OAI-PMH to share data with e.g. discovery services. Now OAI has been extended to support object reuse and exchange, which enables the easy movement of data between different types of repository software, giving substance to the concept of 'open repositories'. More recently we have seen the emergence of large-scale storage devices based on open source software, leading to the term 'open storage'.

Using open storage averts the need for a repository layer to access first-class objects – these are objects that can be addressed directly – where first-class objects include metadata files which point to other first-class objects (such as an ORE representation). We can now begin to realize situations where an institution can exploit the resulting flexibility of repository services and storage: multiple repository softwares can run over a single set of digital objects; in turn these digital objects can be distributed and/or replicated over many open storage platforms.

Being able to select storage enables platforms with error checking and correction functions to be chosen, such as parity (as found in RAID disc array systems), bit checking – a method to verify that data bits have not become corrupted or “switched” – self-recovery and easy expansion. Ordinarily, for economic reasons repositories might not have use of these more resilient storage platforms, but they may become viable for preservation services aimed at multiple repositories.

Early adopters of open storage include Sun Microsystems, which is developing large-scale open source storage platforms, including the STK5800 (codenamed Honeycomb). By focusing on object storage rather than file storage the Honeycomb server provides a resilient storage mechanism with a built-in metadata layer. The metadata layer provides a key component in open storage where objects are given an identifier. For repositories using open storage, there are two scenarios:

1. The repository creates a unique identifier (UID) and URL for an object and the storage platform has to know how to retrieve this object given this identifier.

2. The storage platform creates the UID and/or URL and passes this to the repository on successful creation of the object.

We envisage that both will need to be supported; the first is suited for offline storage mechanisms, whereas the second can be used for cloud and Web 2.0 storage mechanisms.

Aligning with the Web architecture

Three architectural bases of the Web are identification, interaction and formats (Jacobs and Walsh, 2004). It is notable how Web 2.0 applications are designed to be more consistent with the Web architecture than previous-generation Web applications. ORE, for example, with its use of URIs for aggregate resource maps as well as individual objects, opens up new forms of interaction for repository data and extends OAI to conform with Web architectural principles.

We can recognize the growing prevalence of these features, particularly in the number of available APIs. Major services on the Web, such as Google Maps, deploy their own simple APIs. An example within the repository community is SWORD (Simple Web-service Offering Repository Deposit), and open storage platforms such as Sun's STK5800 and the Amazon Simple Storage Service (S3) can similarly be accessed by simple, if different, APIs. To take advantage of open storage, repositories have to be able to talk to these services through these APIs.

An extra feature of STK5800 is Storage Beans, programming code that enables developers to create applications to run on the platform. This is helpful when objects and data need to be manipulated without removing them from the archive.

There is a temptation to try and create standards for methods of communication between applications, especially as in the cases below where the range of potential applications that we may want to work with can be identified. At this stage it appears inevitable that we will have to be adaptable and work with the continuing proliferation of APIs.

Application examples

Storage management

Open repository platforms, which are essentially a set of user and machine interfaces to a built-in storage or database application, are starting to abstract their storage layers to provide flexibility in choice of storage approaches. Increasingly repositories are seen, from a technical angle, as part of a data flow, rather than simply a data destination, and the input and output of data from repositories is supported by applications or interfaces called 'plugins', which can be developed and shared independently without having to modify the core repository software. Typical examples include import

and export of different metadata and reference formats, transfer of XML records, RSS feeds, or data for timelines (Figure 1). EPrints, from version 3.0, is a prominent example of this approach.

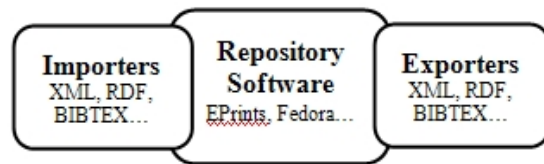


Figure 1: Plugin applications for EPrints prepare data formats for import to, export from, repositories

Adopting the same approach, Preserv 2 is working with the JISC Common Repository/Resource Interface Group (CRIG) and the EPrints technical team to develop a set of expandable plugins to interface EPrints with many types of storage including online and open storage platforms. In addition, EPrints provides a scriptable Storage Controller allowing more than one plug-in to be used to send objects to different storage destinations (Figure 2) based, for example, on the properties of the object or on related metadata. By allowing more than one plugin to be used concurrently it is possible for a plugin to be used specifically for the purposes of long-term preservation services.

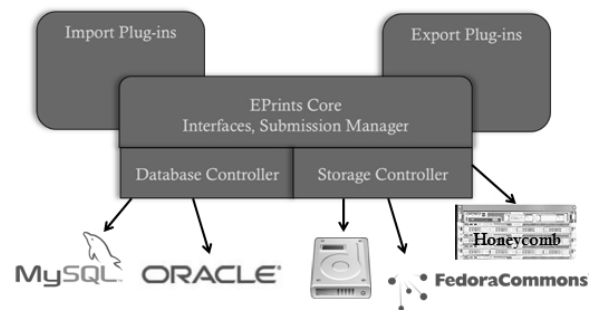


Figure 2: Storage controller, as implemented for EPrints software, enables selected plugins to interface with chosen storage

EPrints is not the only platform developing this sort of architecture. The Akubra project is looking at pluggable low-level storage for Fedora repository software.

Format services

If storage is intended to be a 'passive' preservation approach, in that the aim is to keep the object unchanged, a more active approach is required to ensure that an object remains usable. This requires identification of the format of a digital object and an assessment of the risk posed by that format.

Digital objects are produced, in one form or another, using application programs such as word processors and other tools. These objects are encoded with information to represent characters, layout and other features. The

rules of the encoding are defined by the chosen format of the object. Applications are often closely tied to formats. If applications and formats can change over time, it follows that some risk becoming obsolete – if an application is superseded or becomes unavailable it may not be possible to open objects that were created with that application. This is why formats are a primary focus for preservation actions. The risk to a format can be monitored and might depend on several factors, such as the status of the originating application, or the availability of other tools or viewers capable of opening the format. In some cases objects in formats found to be at-risk may be transformed, or migrated, to alternative formats.

It can be seen from this description that preservation methods affecting formats can be classified in three stages:

- Format identification and characterization (which format?)
- Preservation planning and technology watch (format risk and implications)
- Preservation action, migration, etc. (what to do with the format)

Format-based services tend to be *ad hoc* processes for which some tools are available but which few systems use in a coordinated manner. Currently none of the repository platforms offer support for these tasks beyond basic file format identification using the file extension. Such preservation services can either be performed at the repository management level, or by a trusted third-party service provider. Preserv 2 is working on supporting format services in the cloud alongside open storage, transforming open storage into smart storage. The types of preservation services we are addressing here include file format identification (more than simple extension), risk analysis, and location and invocation of migration tools. All of these require interaction with the repository and access to repository policies. This introduces the need for messaging between the service and the repository, which we address in relation to the services outlined.

Our starting point for this work on smart storage architectures takes existing preservation tools such as PRONOM-DROID (PRONOM [2] is an online registry of technical information, such as file format signatures; DROID [3] is a downloadable file format identification tool that applies these signatures) from The National Archives (UK). In the first phase of Preserv, DROID was implemented as part of a Web service, automatically uploading files from repositories for classification (Brody *et al.* 2007). This uses a lot of bandwidth for large objects, however, and DROID can also become quite processor-intensive. Thus placing this tool alongside storage can decrease the load and bandwidth requirement on the repository while providing most benefit.

Figure 3 shows the implementation of DROID within a smart storage environment. DROID is unchanged from the version distributed by TNA, but three interfaces enable it to interact with an open storage platform and a repository, in this case based on EPrints, which has minor schema changes so that it can accept the metadata generated by DROID.

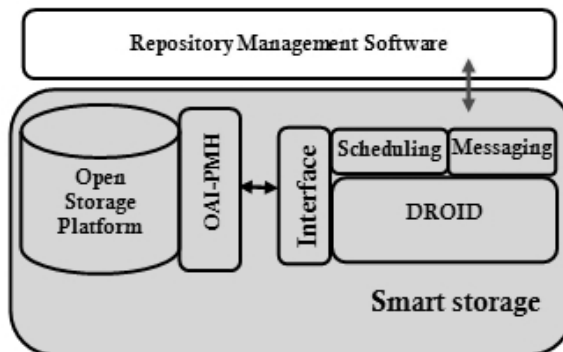


Figure 3: DROID (Digital Record Object Identification) within a smart storage arrangement

The first interface invoked is scheduling, which controls when an update needs to be performed. Preserv 2 has developed a scheduling service based on the Apple iCal calendar format. This interface can thus be controlled directly by the repository by a default repeating event or by a synchronized desktop calendar client. This provides a powerful scheduling service with many clients already available that can read and interpret the files so that both past and future events can be reviewed. In this case the controller around DROID will write the output log into the scheduled event in a log file-type format.

It is anticipated the scheduler will invoke actions based on the results of scanning by DROID allied to decision-making tools that use intelligence from planning and technology watch tools, such as the Plato [4] preservation planning tool from the EC-funded Planets [5] project.

An OAI-PMH interface to open storage discovers the latest objects to have been deposited and which are ready for format classification. Using OAI-PMH is one example of an interface to DROID that can perform this function, but it could also be performed by simpler RSS or Atom-based methods. This interface has since been expanded, again alongside work being done with EPrints, to allow export of OAI-ORE resource maps in both RDF and Atom formats (using the new ORE `rem_rdf` and `rem_atom` datatypes, respectively).

Once new content is discovered a simple controller (not shown in Figure 3) feeds relevant information to DROID, which performs the classifications. At this stage the scheduler is updated and the results are fed to any subscribers, currently by pushing into EPrints.

As a final note on Figure 3 it can be seen that these services and interfaces have been encapsulated within a

smart storage box. Each service has been implemented as Java code and each is able to run alongside the services that are managing the storage API and bit checking.

This implementation provides an early indication of how a decoupled service will need to interface with a range of services and repository management softwares. The simplest method encourages the use of XML and/or RDF for call and callback to and from services. If callback is to happen dynamically between the repository and smart storage, a level of trust needs to be established with this service, and simple HTTP authentication will be required in future releases. A key feature is that all services use RESTful methods for communicating, thus maintaining consistency with the Web architecture, enabling easy plug-ability of new or existing services to a repository.

Further work

Further services are being developed that will be able to interface with representation information registries (Brown 2008) such as PRONOM, which expose information for use by digital preservation services. PRONOM is being expanded as part of Preserv 2 and the EC-funded Planets project to include authoritative information on format risk. Alongside format information a user/agent will then be able to request a risk score relating to a format. This score will be calculated based on several factors each of which has a number of step-based scoring levels, e.g. number of tools available to edit the format.

The Plato preservation tool from the Planets project offers another, in this case user-directed, way of classifying format risks based on specified requirements. The importance of such an approach is that it can take into account the significant properties or particular use cases of a digital object (Knight 2008). Properties of an object that might be considered significant can vary depending who specifies them. Creators, repository managers, research funders in the case of scholarly work, and preservation service providers, can each bring a different view to the features of a digital object that have to be maintained to serve the original purpose.

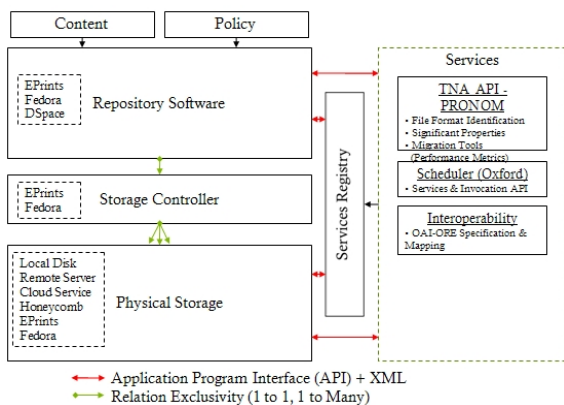


Figure 4: Storage-services based model of Preserv 2 development programme

A more complete picture of how the smart storage approach outlined here fits into the broader programme of Preserv 2 is shown in Figure 4.

Summary

We can place our concept of smart storage within a range of storage approaches and identify a progression:

1. binary stream
2. file system - need to store multiple streams with permissions
3. content addressable - adds content validation and object identifiers, metadata required to locate an object
4. open - adds error correction and recovery, places processing close to storage, solves some bandwidth problems
5. smart - opens up the close-to-storage approach for application development, transition to 'cloud' storage

We also begin to see how smart storage can address the storage problems we encounter:

1. "Billion file" issue - technical scalability of file systems (Wood 2008)
2. Retrieval/indexing - how to locate an item
 - a simple hierarchy is no longer sufficient (RDF maps needed)
 - expectation of Google-style accessibility
 - indexes can themselves require significant storage/processing
3. File integrity - checking, validation, recovery
 - backup as an approach does not scale
 - soft errors become significant
 - bandwidth limits speed of checking, recovery and replication
4. Security/preservation - need for more extensive metadata
 - layered, orthogonal functions over basic storage
5. Application scalability/longevity
 - need to decouple components (Web services or plugins approach, for example)
 - but some functions are bandwidth-hungry, so we need balanced storage/processing at the bottom level
 - use of platform independence (Java, standard APIs) so a "storage bean" can migrate across nodes
 - tightly-coupled Honeycomb is not the only approach, SRB/IRODS is looser
 - with OAI-ORE objects can migrate too
 - very "cloud"-y

- heterogeneous environment - storage policy for different applications/media types, delivery modes

The emergence of this preliminary but flexible framework for managing data from repositories, and the convergence of preservation tools and services, provides the opportunity to reexamine the curation lifecycle, which is being challenged by sharply growing volumes of digital data. The trick will be to identify those traditional approaches that continue to have value, and to adapt and reposition these within the new framework, typically within software. Openness, in its various forms, the ability to move data freely and easily, needs to be supplemented by decision-making that can be automated based on the supplied intelligence and information. In this way, open storage can become 'smarter'.

References

American-Scientist-Open-Access-Forum, 2008a, Convergent IR Deposit Mandates vs. Divergent CR Deposit Mandates, from 25 July 2008

<http://tiny.cc/yM017>

or see <http://amsci-forum.amsci.org/archives/American-Scientist-Open-Access-Forum.html>

American-Scientist-Open-Access-Forum, 2008b, The OA Deposit-Fee Kerfuffle: APA's Not Responsible; NIH Is, see Harnad, S., July 17, and Hitchcock, S., July 18

<http://tiny.cc/YlWdI>

or see <http://amsci-forum.amsci.org/archives/American-Scientist-Open-Access-Forum.html>

Brody, T., Carr, L., Hey, J. M. N., Brown, A. and Hitchcock, S., 2007, PRONOM-ROAR: Adding Format Profiles to a Repository Registry to Inform Preservation Services, *International Journal of Digital Curation*, Vol. 2, No. 2, November

<http://www.ijdc.net/ijdc/article/view/53/47>

Brown, A., 2008, Representation Information Registries, Planets project, White Paper, 29 January

http://www.planets-project.eu/docs/reports/Planets_PC3-D7_RepInformationRegistries.pdf

Hitchcock, S., Brody, T., Hey J. M. N. and Carr, L., 2007, Survey of repository preservation policy and activity, Preserv project, January

<http://preserv.eprints.org/papers/survey/survey-results.html>

Jacobs, I. and Walsh, N. (eds), 2004, Architecture of the World Wide Web, Volume One, W3C Recommendation, 15 December

<http://www.w3.org/TR/webarch/>

Knight, G., 2008, Framework for the definition of significant properties, version: V1, AHDS, InSPECT Project Document, 5 February

<http://www.significantproperties.org.uk/documents/wp33-propertiesreport-v1.pdf>

Lagoze, C. and Van de Sompel, H. (eds), 2008, ORE Specification and User Guide - Table of Contents, 2 June 2008

<http://www.openarchives.org/ore/toc>

Moore, R., 2008, Towards a Theory of Digital Preservation, *International Journal of Digital Curation*, Vol. 3, No. 1

<http://www.ijdc.net/ijdc/article/view/63/82>

RSP, 2008, Commercial Repository Solutions, Repositories Support Project

<http://www.rsp.ac.uk/pubs/briefingpapers-docs/technical-commercial-solutions.pdf>

Wood, C., 2008, The Billion File Problem And other archive issues, *Sun Preservation and Archiving Special Interest Group (PASIG) meeting*, San Francisco, May 27-29, 2008

http://events-at-sun.com/pasig_spring/presentations/ChrisWood_Massive_Archive.pdf

Links

[1] Preserv 2 project <http://preserv.eprints.org/>

[2] Online registry of technical information, PRONOM <http://www.nationalarchives.gov.uk/pronom/>

[3] DROID (Digital Record Object Identification) <http://droid.sourceforge.net/wiki/index.php/Introduction>

[4] Plato - Preservation Planning Tool <http://www.ifs.tuwien.ac.at/dp/plato/>

[5] Planets - Preservation and Long-term Access through NETworked Services <http://www.planets-project.eu/>