Chapter 2

Design and Architecture of Digital Libraries

Hussein Suleman

Introduction

Digital Library Systems (DLSes) are software systems that support the operation of a digital library. As software systems, they are designed primarily to meet the needs of the target community using current best practices in software design and architecture. Digital libraries, like other disciplines, also asserts a set of design constraints that then affect the architectural choices for these digital library systems. Key constraints include: generality, usability by different communities, interoperability, extensibility, preservation and scalability. Individually, these are not unique to DLSes, but together they provide a framework for the development of specific DL architectures.

The DELOS Digital Library Manifesto (Candela, et al, 2007) defines three actors in the architectural space of DLSes. The Digital Library System is the software system that manages data and provides services to users. The Digital Library focuses on the collection, users, processes and services; with a DLS as one of its operational systems. Finally, the Digital Library Management System (DLMS) is responsible for the management of the DLS, for example instantiation of collections and services.

This chapter focuses on the DLS and, to a lesser degree, the DLMS. Core design considerations are first presented, followed by how these principles are realised in modern reusable and custom-built DLSes. The next section deals with how these individual systems are interconnected into larger networked DLSes, exemplified by international projects such as NDLTD. Scalability – how to deal with increasing sizes of data and increasing numbers of service requests – is then discussed. Finally, the chapter ends with research directions and a case study of an architecture designed for the developing world.

Core Design Considerations

Core Components

Most digital library systems contain 3 main components: a digital object store, a metadata store and a suite of services to manage and provide access to the other 2 components. These are depicted in Figure 1.

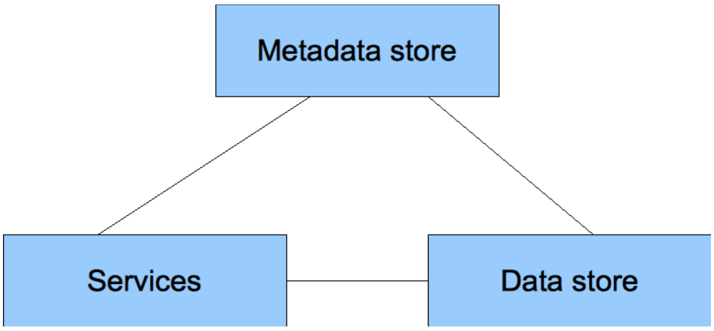


Figure 1. Three main components of a DLS architecture

The digital object store and metadata store are typically implemented using a combination of filesystems, databases, triple stores, etc.

Services are provided by applications that execute locally and via remote interfaces such as Web-based interfaces – the exact mapping of services to applications varies across architectures. Typical services provided by a DLS include: search, browse, submit, annotate, manage, copy, authorize, import, export, link, filter and visualize (Gonçalves, et al, 2004). Examples of such systems are presented in the subsequent sections of this chapter.

Service Oriented Architecture

Service Oriented Architecture (SOA) is a popular paradigm in DLS architecture. Dienst (Lagoze & Davis, 1995) was one of the earliest examples of a designed distributed DLS with a strong emphasis on services and components. It was based on the Kahn/Wilensky Architecture (Kahn & Wilensky, 2006), which defined a set of simple primitives for abstract access to a DLS, and the Warwick Framework (Lagoze, 1996), which defined an abstract metadata container mechanism for genericity and multiplicity in metadata management. The influence of these frameworks has been carried forward into the Fedora repository toolkit (Payette & Lagoze, 1998). Fedora provides only a carefully managed repository with APIs and no user interfaces; the intention is that other systems will build on Fedora's strong foundation.

The emergence of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) (Lagoze & Van de Sompel, 2001) led to more support for the idea of communicating components in a service-oriented architecture. Building on the success of OAI-PMH, the Open Digital Libraries (Suleman & Fox, 2002) and OpenDLib (Castelli & Pagano, 2002) projects attempted to define service interfaces for more than just harvesting.

As the number of components increased, management became crucial and two approaches were proposed. Diligent (Candela, et al, 2006) extended OpenDLib to a grid environment to exploit the facilities provided by grid computing infrastructure. In contrast, Blox (Eyambe & Suleman, 2004) developed a minimal DLMS based also on SOA.

All of these project have illustrated the viability of components interconnected using well-defined protocols as the core architecture of a DLS. Increasingly this is becoming the norm in DLS architecture as it is in many other domains. As examples, Greenstone 3 (Bainbridge, et al, 2004) separates major components internally using SOA and the core of the Europeana system is based on externally accessible services (Concordia, et al., 2009).

Digital Library Systems

Custom-built Systems

Many early digital library systems were designed to meet a specific goal and the software was considered to be specific to that goal. ArXiv.org is a central archive of preprints and postprints in the extended Physics research community. The architecture of the system, the metadata and data it stores and the services it provides to its users are all driven completely by the needs of only its user community. The same holds for non-profit DLSes like the ACM Digital Library (www.acm.org/dl) and for-profit DLSes like SpringerLink. While all of these systems have been influenced by best practices in the architecture of DLSes, this is only noticeable in the external interfaces. For example, global identifier schemes for persistent linking is available on many such systems.

Institutional Repository Toolkits

The Open Access Movement has supported the design of reusable DLSes, as the use of a standard institutional repository tool is one part of an Open Access solution for an institution (Harnad, et al, 2004). The most popular tools to serve as the support software for an institutional repository are currently EPrints (www.eprints.org) and DSpace (www.dspace.org). OpenDOAR (www.opendoar.org), a registry of Open Access repositories, lists 2160 repositories as of 12 December 2011. 1739 of these repositories each use one of 80 different named DLSes. Only EPrints and DSpace have more than 100 instances each. In fact, only 16 DLSes have more than 10 instances each, with a large majority of the DLSes having only a single instance. Thus, in practice, there are both large numbers of repositories with custom software solutions and large numbers of repositories using standard tools. Many of the systems in the former category were designed for specific projects and later generalised.

Both DSpace and EPrints, which together account for approximately half of the systems listed on OpenDoar, offer the following features:

- browse, search and submission services;
- basic workflow management for submission, especially editing of metadata and accepting/rejecting submissions;
- network-oriented installation (i.e., installation without a live network connection is not recommended);
- customisable Web interfaces;
- external import and export functions; and
- interoperability interfaces such as OAI-PMH (Lagoze & Van de Sompel, 2001) and SWORD (Allinson, 2010).

A major difference is that DSpace can only use qualified Dublin Core as its metadata format while EPrints allows for the definition of arbitrary metadata formats.

Besides these systems, other repository toolkits have been developed with different design goals. Invenio (invenio-software.org) from CERN provides a large suite of very flexible services but installation and configuration are not as simple as DSpace/EPrints. Fedora (Payete and Lagoze, 1998), in contrast, provides users with a strong foundation repository but does not come bundled with any end-user interfaces or workflow management systms. Fez (fez.library.uq.edu.au) is an institutional repository tool built on top of Fedora but its small user base means that installation and support are not on par with DSpace/EPrints.

Commercial offerings attempt to deal with some of these problems, which appear to be largely about software configuration and management. Zentity (research.microsoft.com/en-us/projects/zentity) is a Microsoft toolkit that can be used to create a general-purpose repository with visualization as a core service. Hosted solutions are more popular: Digital Commons (digitalcommons.bepress.com) from BEPress allows repository managers to completely avoid the problems of software systems by hosting their collections and services remotely and dealing only with the content-related aspects.

The remote hosting of collections occurs also in the Open Source community, where one institution may host the DLS of another that may not have the hardware or personnel to do so. This model is used in the South African National ETD Project (Webley, et al, 2011), where smaller institutions have hosted collections at a central site.

Cultural Heritage and Educational Resources

Systems for cultural heritage preservation use DLSes to preserve and provide access to digital representations of artefacts. These DLSes differ from the other repository toolkits because they offer specific preservation and discovery services for highly specialised collections of data.

The Bleek and Lloyd collection (Suleman, 2007) of Bushman stories was designed for distribution and access without a network and can be viewed off a DVD-ROM using a standard Web browser. The Digital Assets Repository at Bibliotheca Alexandria (Mikhail, et al, 2011) was designed for large scale storage of digital objects using a flexible, modular and scalable design. Besides such custom-built solutions, the Greenstone Digital Library (Witten, et al, 2001) toolkit allows end-users to easily create their own indexed collections with search and browse functionality. The emphasis of Greenstone's design has been on universal applicability and minimal resource use.

Digital library systems have also been used for educational resources. The National STEM Digital Library (nsdl.org) is a large and interconnected system of repositories to gather and provide easy access to Science, Technology, Engineering and Mathematics resources for educators and learners. Unlike the previous systems, the architecture of NSDL is inherently distributed – the motivation for this and similar large-scale systems is presented in the next section.

Central vs. Distributed Architectures

Motivation for Distribution

A central DLS stores all its digital objects and metadata and provides services from a single hardware system in a single location. In contrast, a distributed DLS may store digital objects and/or metadata in multiple locations and/or may provide services from multiple locations. Popular DLS tools, such as DSpace and EPrints, create a central DLS but include services for interconnection into larger networked infrastructures.

Distributed DLSes are desirable for a number of reasons, including that:

- central DLSes are resource-intensive, while in distributed DLSes the costs are shared among the distributed partners;
- different collections of digital objects and metadata usually belong to different organisations and a distributed system maintains the links between organisations and their collections; and
- services may be provided by the most appropriate service providers rather than by the data owners by default.

While distributed DLSes have clear benefits, end-users need cross-archive discovery and access services as they cannot navigate the space of thousands of collections to find relevant resources. The simplest approach to provide such cross-archive services is a Web search engine. Simply by virtue of having a visible online presence on the WWW, a repository will be indexed by search engines that crawl the Web regularly, e.g., Google and Bing. Provided that such search engines are able to distinguish high quality digital resources from less useful websites, some useful services may be provided. Google Scholar and Microsoft Academic Search are examples of services based on crawled data. In both cases, the indices are of a relatively high quality but neither uses the more complete metadata found in DLSes to provide higher-order services such as alerts or structured browsing and neither can provide access to well-defined subsets of digital objects, such as theses. Such higher-order and focused services are arguably best provided by direct interoperability among DLSes; most recently this has been accomplished through federation or harvesting.

Federation

Federation refers to services that access distributed sites on-demand in order to satisfy a specific request. The most popular form of federation is federated search, where a query sent to a DLS is forwarded to one or more remote sites. The results from the remote sites may then be merged into a single result set using result fusion (Shokouhi, 2007). The set of remote sites to query also can be optimised by eliminating those sites that probably do not contain relevant results – this is called source selection.

In practice, federated search requires that a DLS support a remote search protocol such as Z39.50 or the newer Search/Retrieval via URL (SRU) (www.loc.gov/standards/sru). SRU is a client-server protocol where the client sends URL-encoded query parameters to the server and receives an XML-encoded list of records as a result.

Early experiments with federated search were partially successful but the reliance of a DLS on multiple remote sites led to unreliable operation over time, with increasing unreliability with greater numbers of sites. This inability of federation to scale and remain stable over time led to harvesting as an alternative method of creating distributed DLSes (Suleman & Fox, 2003).

Harvesting and Open Archives

Harvesting is the periodic transfer of data from one machine to another. In the digital libraries community, the data being transferred is usually metadata that has been updated since the last harvest. Harvesting has been suggested as an alternative to federation because a DLS that collects all the metadata in one location does not need to contact each remote site on every query. The disadvantage is that all metadata has to be stored in one location – however the digital objects are usually not harvested, the central metadata store is easily replicated and metadata does not use much storage relative to the digital objects themselves.

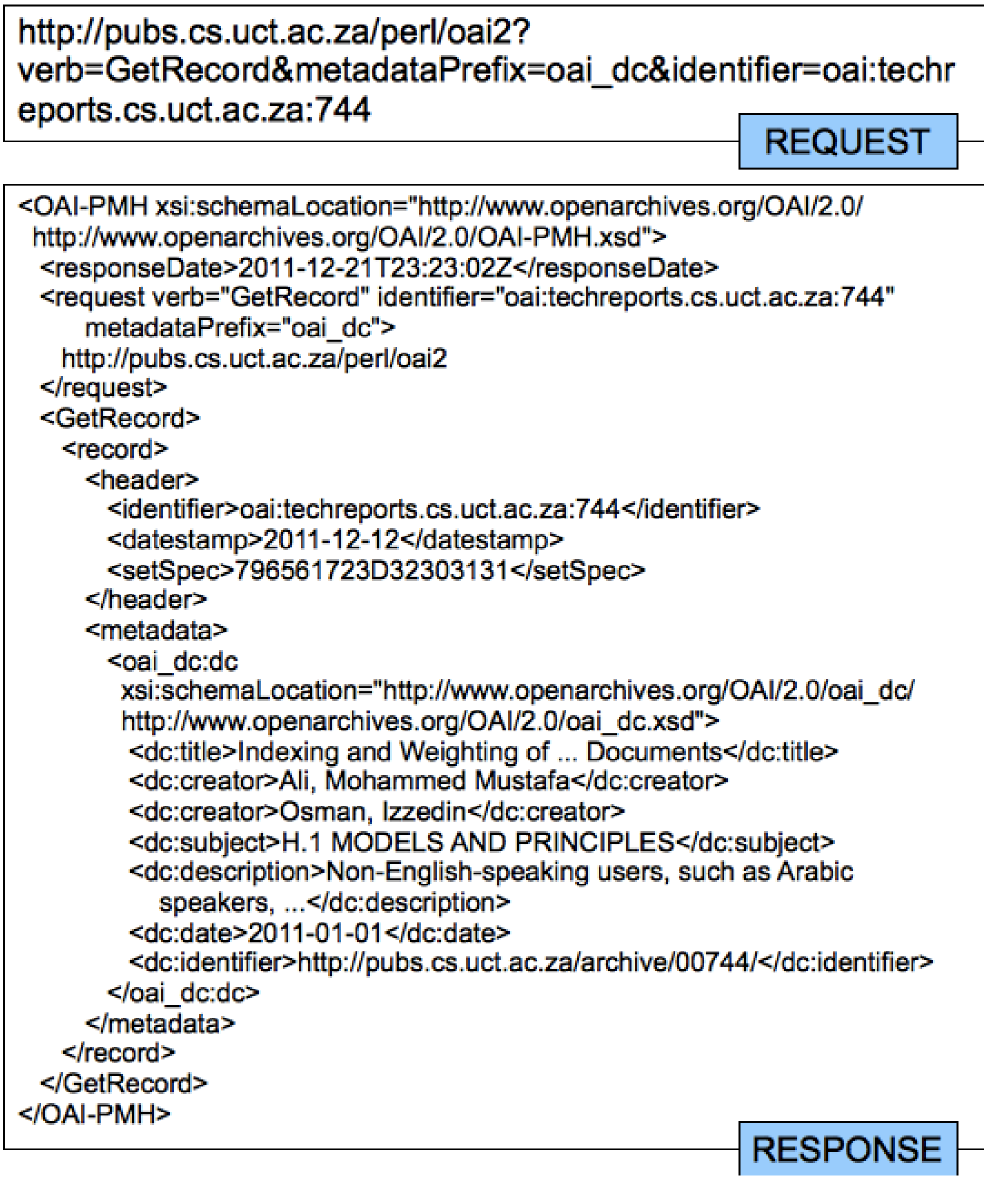
The Open Archives Initiative (OAI) developed the Protocol for Metadata Harvesting (PMH) (Lagoze, et al, 2002a) as a mechanism for interconnecting distributed repositories using the harvesting approach. This is a client-server Web-based protocol, where requests are URL-encoded and responses are contained in well-defined XML documents. There are 6 possible requests in the OAI-PMH, as listed below:

- **Identify** returns a description of the repository, including such information as the administrator's email address, service endpoint URL and name of repository.
- **ListMetadataFormats** returns a list of the metadata formats in which records may be disseminated.
- **ListSets** returns a list of subsets of the repository that may be harvested instead of the entire collection.
- **ListIdentifiers** returns a list of record headers, where each header contains the identifier of a record, its update date and the list of sets it belongs to.
- **GetRecord** returns a single complete metadata record for a specified identifier and metadata format, including header information and optional meta-metadata, such as provenance data.
- **ListRecords** is a combination of the ListIdentifiers and GetRecord requests, where the repository sends back a list of complete records instead of just their headers.

OAI-PMH relies on datestamps for incremental harvesting. A ListIdentifiers or ListRecords request can specify a 'from' parameter that indicates the earliest datestamp of records to be returned. If all records are datestamped on accession or modification, a client can then harvest only the updated and new records on each subsequent harvest after the first. Since this is a stateless solution, multiple clients also can independently harvest at different times.

There is, however, one element of state in the protocol in the form of a resumptionToken. This is a special parameter that is returned in one of the 'List' responses to indicate that there is more data available and that the client should send this token back to request the next installment of the data. This mechanism allows servers to send batches of records to clients without creating XML documents of an unmanageable size.

Figure 2 illustrates a typical OAI request and response.



```
http://pubs.cs.uct.ac.za/perl/oai2?
verb=GetRecord&metadataPrefix=oai_dc&identifier=oai:techr
eports.cs.uct.ac.za:744
```

```
<OAI-PMH xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
 http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2011-12-21T23:23:02Z</responseDate>
  <request verb="GetRecord" identifier="oai:techreports.cs.uct.ac.za:744"
       metadataPrefix="oai_dc">
    http://pubs.cs.uct.ac.za/perl/oai2
  </request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:techreports.cs.uct.ac.za:744</identifier>
        <datestamp>2011-12-12</datestamp>
        <setSpec>796561723D32303131</setSpec>
      </header>
      <metadata>
        <oai_dc:dc
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
         http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title>Indexing and Weighting of ... Documents</dc:title>
          <dc:creator>Ali, Mohammed Mustafa</dc:creator>
          <dc:creator>Osman, Izzedin</dc:creator>
          <dc:subject>H.1 MODELS AND PRINCIPLES</dc:subject>
          <dc:description>Non-English-speaking users, such as Arabic
              speakers, ...</dc:description>
          <dc:date>2011-01-01</dc:date>
          <dc:identifier>http://pubs.cs.uct.ac.za/archive/00744/</dc:identifier>
        </oai_dc:dc>
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>
```

Figure 2. OAI-PMH Request and Response

Examples of distributed architectures

National STEM Digital Library

The National Science Technology Engineering and Mathematics (STEM) Digital Library (NSDL) is an American project to organise and provide access to teaching and learning resources in the STEM disciplines, spanning K-12 primary/secondary schools as well as university education (nsdl.org). NSDL stores mostly metadata but has also indexed full-text content where possible. The metadata used was either standard unqualified Dublin Core or an NSDL-specific variant (Lagoze, et al, 2002b), to ensure that participation in NSDL had few barriers.

NSDL's technical architecture focuses on a central metadata repository that stores metadata from all its distributed partner repositories. The metadata in this repository is in turn shared with service providers, such as a local search service, via an OAI-PMH interface. Ingest into the repository takes place using one of 3 mechanisms: direct entry into the database; harvesting using OAI-PMH; or gathering from the Web, a form of Web-crawling. This 'spectrum of interoperability' was decided on to try to maximise participation – while harvesting was considered the normative interoperability mechanism, it was expected that harvesting is not as low a barrier to interoperability as is needed for repository sites with few human resources to set up and operate OAI-PMH interfaces. As of 2006, NSDL contained 1.2 million records and regularly harvested from 85 OAI-PMH servers (Lagoze, et al, 2006).

Europeana

Europeana is a cultural heritage digital library project with an emphasis on gathering the heritage of different European communities into a single portal for easy navigation and discovery by end-users (www.europeana.eu). As with other distributed digital libraries, the content is stored at the distributed partners but the metadata is shared with the central portal. This central portal stores metadata as a networked data system based on RDF so that the services can exploit the rich relationships among and within objects and collections (Doerr, et al, 2010). Fundamentally, however, the data is either harvested using OAI-PMH or manually inserted into the central repository using a defined API (Concordia, et al, 2009). In this way, the core data architecture differs from NSDL, but the core networked architecture is the same.

Networked Digital Library of Theses and Dissertations

The Networked Digital Library of Theses and Dissertations operates a Union Catalog of metadata describing Electronic Theses and Dissertations (ETDs) from around the world (union.ndltd.org). Metadata is stored in a central metadata repository in either Dublin Core or ETD-MS formats – ETD-MS is a metadata format specific to ETDs. Like both NSDL and Europeana, metadata is harvested periodically from partner sites using the OAI-PMH. Sites may represent either single institutions (e.g., Virginia Tech), consortia (e.g., OhioLink), country/regional projects (e.g., Australasian Digital Theses) or international collaborations (e.g., OCLC WorldCat). Given that these are overlapping organisations, the Union Catalog contains some repeated records – de-duplication of these is an ongoing challenge.

Services are provided at a higher level by independent service providers obtaining a stream of metadata from the Union Catalog, again using the OAI-PMH. As of 2011, VTLS and Scirus provide discovery interfaces based on the metadata.

The NDLTD Union Catalogue is strongly focused on a single type of resource but agnostic to the source of the metadata and the language and cultural differences in higher education systems around the world. As such, its architectural model is therefore a generic model for international focused DLSes. This is illustrated in Figure 3.

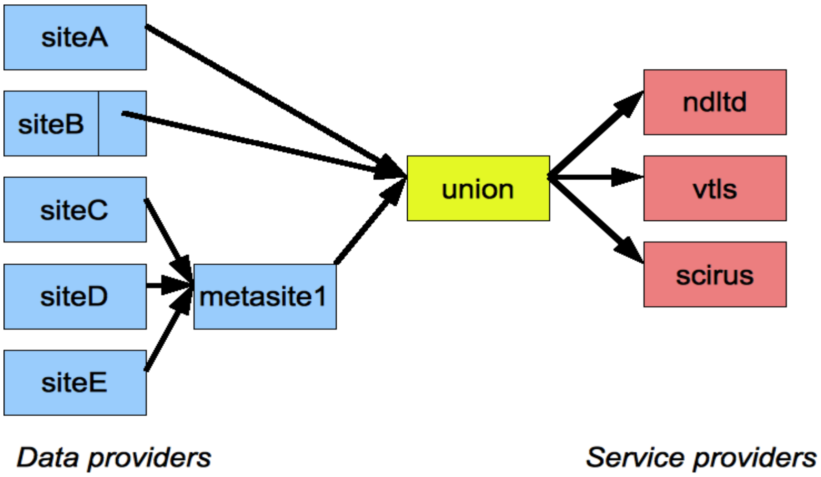


Figure 3. NDLTD data and service provider network

Scalability

DLSes such as the ones presented above have to deal with 3 forms of scalability challenges:

- increasing numbers of service requests (or end users);
- increasing amounts of data and metadata; and
- increasing numbers of repositories.

The first two forms of scalability have been explored recently while the last is gradually emerging as more digital repository systems are created.

Compute Clusters

Clusters of tightly-integrated networked computers may be used for high performance computing or high throughput computing tasks, both of which are necessary as quantities of data and numbers of end-users increase. Suleman, et al (2008) performed experiments using a cluster of computers to provide typical DLS services and illustrated that it is indeed feasible to replicate services on multiple nodes to handle increases in the number of requests. Nakashole & Suleman (2009) showed that indexing of large quantities of data in a DLS can be done effectively in parallel using both fixed cluster nodes and dynamic grid-style nodes.

Cloud/Utility Computing

Utility computing allows for the creation of one or more virtual computers in a remote location; or the creation of arbitrarily large storage systems in remote locations. Any Web-based DLS can exploit such services in its architecture to support scaling of services or data stores.

Utility computing is either a pay-per-use service by commercial providers (e.g., Amazon) or a locally-hosted service with the same interfaces but on a local computing cluster, referred to as a private cloud. The advantage of the former is that a popular service can quickly and relatively cheaply acquire large amounts of resources without having to deal with power and cooling concerns. Private clouds, in contrast, allow an organisation to maintain full control of the hardware and data, where this is potentially an issue. In the DLS space, few projects have adopted utility computing as part of their architectures.

Volunteer Thinking/Crowdsourcing

Volunteer computing is a large-scale processing of data using the idle computational resources of volunteers. This has usually been applied to solve humanitarian, medical or scientific problems that require processing of large quantities of data.

This approach has recently been generalised to volunteer thinking or crowdsourcing, where users are recruited to performs tasks that computers are not able to perform easily, such as image recognition. This has been applied in the context of digital libraries for the creation of high quality metadata and transcription of handwritten text (Oomen & Arroyo, 2011). The advantage over automated techniques is that large numbers of volunteers can be recruited to deal with large quantities of data and repeated processing of the same data ensures high quality in addition to high speed and low cost.

Scalable Storage

Database scalability provides a trivial basis for scaling DLS storage where the metadata and/or data collections reside in databases. In addition, specific techniques may be employed to achieve scalability at higher levels.

The Amazon Web Services (aws.amazon.com) utility computing infrastructure offers multiple services for data storage where the physical details are masked from the system. S3 is a service to store arbitrary amounts of data addressed with a logical identifier scheme. SimpleDB is a service to store indexed tuples and perform queries using a variant of SQL. Both services are provided as Web-based APIs for easy integration into external systems or systems that run on the co-located virtual machine infrastructure (EC2). In the digital library community, DuraCloud is using multiple utility storage providers as a replication platform for preservation of digital objects as a commodity service (www.duraspace.org/duracloud.php).

Instead of centralised storage, systems may opt to use multiple distributed storage systems to achieve scalability, flexibility and extensibility. This is the approach advocated by Storage Resource Broker (SRB) (Baru, et al, 1998) and its successor – iRODS. Both mechanisms create abstractions of remote storage. Some repository toolkits, such as DSpace, support SRB/iRODS as an abstract storage layer.

Case Study: simplyCT

Motivation

SimplyCT is an alternative architecture for digital library systems with a specific emphasis on low resource environments, such as institutions in developing countries. Low resource environments are environments where there are few staff; the staff are not highly skilled in digital library systems; there are few servers and other computers available; the computers are not high-end and are shared among multiple applications; and there is either no network available or the network is slow and unreliable. Taking all the above conditions into account, simplyCT has been proposed as an architectural framework that is potentially more relevant for developing countries.

The basic idea behind simplyCT is to encourage simplicity in all elements of the architecture. This strategy is borne out by the long-term success of Project Gutenberg, which has a stated preference for simple text to guarantee preservation of the data. The initial experiences of NSDL, where OAI-PMH turned out to be too complex, also support the notion that complex DLSes are not likely to be effective without large injections of resources (Lagoze, et al, 2006).

SimplyCT is currently a proposal that is being evaluated experimentally and by direct application in production systems such as the Bleek and Lloyd Collection (lloydbleekcollection.cs.uct.ac.za) of Bushman stories and drawings.

Principles

SimplyCT is based on a set of design principles derived from the experiences of past DLSes. These principles are as follows:

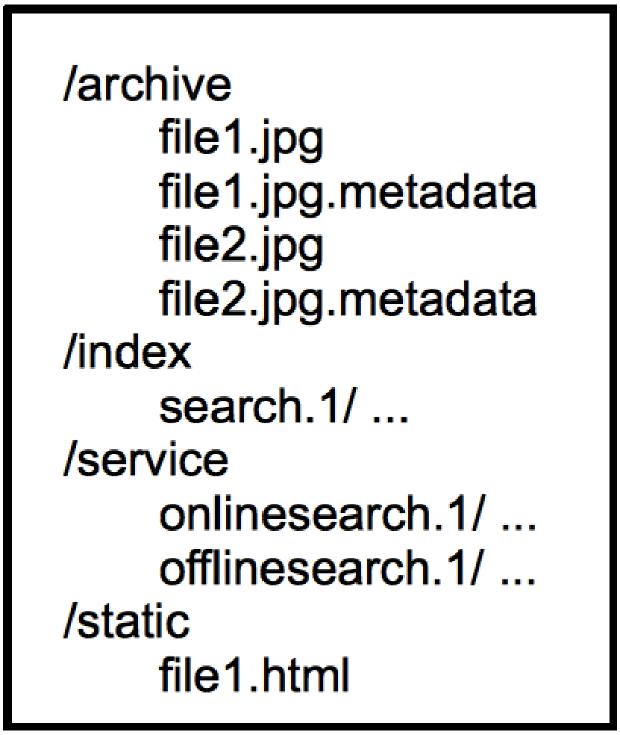
- Minimalism. Only provide the bare minimum amount of infrastructure as additional complexity increases development, extensibility and maintenance costs. This principle argues against highly layered and abstract architectures.
- Do not impose on users. Users may already use the data in particular formats and with specific structures and identifiers. The DLS should mould itself around the data rather than force users to changes structures, formats and identifiers. This principle argues against enforced global identifiers.
- No API. Components should not need to use an API when basic file access is sufficient. This principle argues against Web Services for everything.
- Web or No Web. DLSes should not require the use of a Web interface where users may want to access data by other means. This principle argues against unnecessary layers for users to access data.
- Preservation by Copying. It should be possible to preserve the data and services simply by copying a directory, as this form of data protection is widely understood and can be implemented using a wide range of technological solutions. This principle argues against the use of unnecessary databases.
- Any metadata, objects, services. There should be no constraints on what metadata formats are allowed, what digital objects are allowed or what services are possible. This principle argues against unnecessary restrictions on formats.
- Everything is repeatable. Even basic services should be repeatable so that, for example,

different search services can be provided over core metadata and annotations. This principle argues against fixed services.

- Superimposed information. Metadata should be stored as granular objects with additional layers of meaning specified as superimposed peer sub-collections. This principle argues against complex objects that conflate descriptive metadata with categorical metadata.

Structure

Figure 4 illustrates the file structure of a simplyCT archive.



```
/archive
        file1.jpg
        file1.jpg.metadata
        file2.jpg
        file2.jpg.metadata
/index
        search.1/ ...
/service
        onlinesearch.1/ ...
        offlinesearch.1/ ...
/static
        file1.html
```

Figure 4. simplyCT file layout

'archive' contains the digital objects and metadata files corresponding to each digital object. The metadata files have the same names as the digital objects but are suffixed with '.metadata'. 'index' contains indices for any services that require these, such as search. 'service' contains the code and/or configuration data for each instance of each service. 'static' contains any static files that are part of the user interfaces, like the Home page of the collection.

Toolsets

Some tools have been constructed to manipulate and access data based on the simplyCT framework.

The CALJAX project (Suleman, et al, 2010) produced a Web-based collection manager and AJAX-based search and browse tools for a simplyCT-structured collection. The access services were based on AJAX so that the collection could be copied to a CD/DVD-ROM without any loss in functionality even though there is no longer a Web server intermediary.

The Bonolo project (http://shenzi.cs.uct.ac.za/~honsproj/cgi-bin/view/2011/hammar_robinson.zip/Website/) produced Web-based tools for both management and discovery of simplyCT collections. The two tools were developed to be independent so that the management module could be used to maintain the collection while CALJAX's services are used for access.

Further tools are under development.

Research challenges and trends in DLS architecture

Current research challenges include the following:

- Large data curation refers to the ingest, management, querying and dissemination of large datasets. As datasets become very large, the standard algorithms for handling data change and new algorithms are actively being sought.
- Metadata and system interoperability is still difficult at best. Experience with OAI-PMH has shown that, while it works in some communities, it is not a good general solution. Newer standards like OAI-ORE are significantly more complex than is probably necessary.
- Packaging of DLSes is often a source of frustration because dependencies and configurations are handled as out-of-band activities. Instead, modern packaging systems like FreeBSD ports and Ubuntu packages can provide system-wide management across all software systems. Integrating DLS tools into OSes can result in more widespread use and adoption.
- Cloud services are increasing in popularity. New services can provide annotation and linking capabilities and these should be provided in DLSes. In addition, there is a need for more evidence that cloud services are a good design decision.
- Finally, any distributed or large DLS needs to monitor its metadata for duplicates and quality issues. Services to make such assessments are gradually being improved but deduping is still considered to be a difficult problem.

Summary

In summary, the architecture of a modern digital library system is based on a metadata store, data store and service suite. The service suite typically follows a service oriented architecture, which has been experimentally validated in various systems.

Distributed DLSes are common and interoperability is a key consideration for any system. Most systems are therefore not run in isolation but as one part of a larger networked information system, where scalability adds a new dimension to the architecture. Standards such as OAI-PMH are crucial in enabling this interoperability for such systems.

However, even OAI-PMH has been demonstrated as too complex for some applications. In addressing this, simplyCT has been presented as an alternative architecture for DLSes, based on a set of principles and minimal services. It is hypothesized that an architecture conforming to these principles will stand the test of time and have greater universal applicability than existing designs for DLSes. This shows much promise in early experiments but is yet to be proven by ongoing research!

Bibliography

Allinson, J., Carr, L., Downing, J., Flanders, D. F., Francois, S., Jones, R., Lewis, S., Morrey, M., Robson, G., and Taylor, N. (2010) *SWORD AtomPub Profile version 1.3: Simple Webservice Offering Repository Deposit*. Available http://www.swordapp.org/docs/sword-profile-1.3.html

Bainbridge, D., Don, K. J., Buchanan, G., Witten, I. H., Jones, S., Jones, M., and Barr, M. I. (2004) Dynamic Digital Library Construction and Configuration, in Heery, R., and Lyon, L. (eds): *Research and Advanced Technology for Digital Libraries: ECDL 2004*, Springer, LNCS 3232. doi:10.1007/978-3-540-30230-8_1

Baru, C., Moore, R., Rajasekar, A., and Wan, M. (1998) The SDSC storage resource broker, in *Proceedings of the 1998 Conference of the Centre for Advanced Studies on Collaborative Research*, ACM Press.

Candela, L., Castelli, D., Ioannidis, Y., Koutrika, G., Pagano, P., Ross, S., Schek, H.-J., and Schuldt, H. (2006) *The Digital Library Manifesto*, DELOS, January. Available http://www.delos.info/index.php?option=com_content&task=view&id=345

Candela, L., Castelli, D., Pagano, P., and Simi, M. (2006) OpenDLibG: Extending OpenDLib by Exploiting a gLite Grid Infrastructure, in Gonzalo, J., Thanos, C., Verdejo, M. F., and Carrasco, R. C. (eds): *Research and Advanced Technology for Digital Libraries: ECDL 2006*, Springer, LNCS 4172. doi:10.1007/11863878_1

Castelli, D., and Pagano, P. (2002) OpenDLib: A Digital Library Service System, in Agosti, M., and Thanos, C. (eds): *Research and Advanced Technology for Digital Libraries*, Springer, LNCS 2458. doi:10.1007/3-540-45747-X_22

Concordia, C., Gradmann, S., and Siebinga, S. (2009) Not (just) a Repository, nor (just) a Digital Library, nor (just) a Portal: A Portrait of Europeana as an API, in *Proceedings of World Library and Information Congress: 75th IFLA General Conference and Council*, IFLA, 23-27 August, Milan, Italy. Available http://www.ifla.org/files/hq/papers/ifla75/193-concordia-en.pdf

Doerr, M., Gradmann,S., Hennicke, S., Isaac, A., Meghini, C., and Van de Sompel, H. (2010) The Europeana Data Model (EDM), in *Proceedings of World Library and Information Congress: 76th IFLA General Conference and Council*, IFLA, 10-15 August, Gothenburg, Sweden. Available www.ifla.org/files/hq/papers/ifla76/149-doerr-en.pdf

Eyambe, L., and Suleman, H. (2004) A Digital Library Component Assembly Environment, in Marsden, G., Kotzé, P., and Adesina-Ojo, A. (eds): *Proceedings of SAICSIT 2004*, 4-6 October, Stellenbosch, ACM Press, 15-22.

Gonçalves, M. A., Fox, E. A., Watson, L. T., and Kipp, N. A. (2004) Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries, *ACM Transactions on Information Systems*, ACM, **22** (2).

Harnad, S., Brody, T., Vallières, F., Carr, L., Hitchcock, S., Gingras, Y., Oppenheim,C., Stamerjohans, H., and Hilf, E. R. (2004) The Access/Impact Problem and the Green and Gold Roads to Open Access, *Serial Review*, Elsevier, **30** (4), 310-314. doi:10.1016/j.serrev.2004.09.013

Kahn, R., and Wilensky, R. (2006) A framework for distributed digital object services, *Intl Journal on Digital Libraries*, Springer, **6** (2), 115-123. doi:10.1007/s00799-005-0128-x

Lagoze, C. (1996) The Warwick Framework: A Container Architecture for Diverse Sets of Metadata, *D-Lib Magazine*, July/August. Available http://www.dlib.org/dlib/july96/lagoze/07lagoze.html

Lagoze, C., and Davis, J. R. (1995) Dienst: an architecture for distributed document libraries, *Communications of the ACM*, ACM, **38** (4), 47. doi:10.1145/205323.205331

Lagoze, C., and Van de Sompel, H. (2001) The open archives initiative: building a low-barrier interoperability framework, in Fox, E. A., and Borgman, C. L. (eds): *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*, ACM, 54-62. doi:10.1145/379437.379449

Lagoze, C., Van de Sompel, H., Nelson, M., and Warner, S., (2002a) *The Open Archives Initiative Protocol for Metadata Harvesting, Version 2.0*, Open Archives Initiative. Available http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm

Lagoze, C., Hoehn, W., Millman, D., Arms, W., Gan, S., Hillman, D., Ingram, C., Krafft, D., Marisa, R., Phipps, J., Saylor, J., Terrizzi, C., Allan, J., Guzman-Lara, S., and Kalt, T. (2002b) Core Services in the Architecture of the National Science Digital Library (NSDL), in Hersh, W., and Marchionini, G. (eds): *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, ACM, 201-209. doi:10.1145/544220.544264

Lagoze, C., Krafft, D., Cornwell, T., Dushay, N., Eckstrom, D., and Saylor, J. (2006) Metadata aggregation and 'automated digital libraries': a retrospective on the NSDL experience, in Nelson, M. L., and Marshall, C. C. (eds): *Proceedings of the 6nd ACM/IEEE-CS Joint Conference on Digital Libraries*, ACM, 230-239. doi:10.1145/1141753.1141804

Mikhail, Y., Adly, N., and Nagi, M. (2011) DAR: Institutional Repository Integration in Action, in Gradmann, S., Borri, F., Meghini, C., and Schuldt, H. (eds): *Proceedings of Research and Advanced Technology for Digital Libraries, International Conference on Theory and Practice of Digital Libraries (TPDL 2007)*, 26-28 September, Budapest, Hungary, Springer, LNCS 6966, 348-359. doi:10.1007/978-3-642-24469-8_36

Nakashole, N., and Suleman, H. (2009) A Hybrid Distributed Architecture for Indexing, in Agosti, M., Borbinha, J, Kapidakis, S., Papatheodorou, C., and Tsakonas, G. (eds): *Research and Advanced Technology for Digital Libraries: Proceedings of 13th European Conference (ECDL 2009)*, 27 September – 2 October, Corfu, Greece, Springer, LNCS 5714, 250-260. doi:10.1007/978-3-642-04346-8_25

Oomen, J., and Arroyo, L. (2011) Crowdsourcing in the cultural heritage domain: opportunities and challenges, in *Proceedings of 5th International Conference on Communities & Technologies*, 29 June – 2 July, Brisbane, Australia.

Payette, S., and Lagoze, C. (1998) Flexible and Extensible Digital Object and Repository Architecture (FEDORA), in Nicholaou, C., and Stephanidis, C. (eds): *Research and Advanced Technology for Digital Libraries*, Springer, LNCS 1513. doi:10.1007/3-540-49653-X_4

Shokouhi, M. (2007) Segmentation of Search Engine Results for Effective Data-Fusion, in Amati, G., Carpineto, C., and Romano, G. (eds): *Advances in Information Retrieval*, Springer, LNCS 4425, 185-197. doi:10.1007/978-3-540-71496-5_19

Suleman, H. (2007) Digital Libraries Without Databases: The Bleek and Lloyd Collection, in Kovacs, L., Fuhr, N., and Meghini, C. (eds): *Proceedings of Research and Advanced Technology for Digital Libraries, 11th European Conference (ECDL 2007)*, 16-19 September, Budapest, Hungary, 392-403.

Suleman, H., and Fox, E. A. (2001) A Framework for Building Open Digital Libraries, *D-Lib Magazine*, **7** (12). Available http://www.dlib.org/dlib/december01/suleman/12suleman.html

Suleman, H., and Fox, E. A. (2003) Leveraging OAI Harvesting to Build a Union Catalog, *Library Hi-Tech*, **21** (2), Emerald Publishing, 219-227. doi:10.1108/07378830310479857

Suleman, H., Parker, C., and Omar, M. (2008) Lightweight Component-Based Scalability, *International Journal on Digital Libraries*, **9** (2), Springer, 115-128.

Suleman, H., Bowes, M., Hirst, M., and Subrun, S. (2010) Hybrid Online-Offline Collections, in *Proceedings of Annual Conference of the South African Institute for Computer Scientists and Information Technologists (SAICSIT 2010)*, Bela Bela, South Africa, 11-13 October, ACM Press.

Webley, L., Chipeperekwa, T., and Suleman, H. (2011) Creating a National Electronic Thesis and Dissertation Portal in South Africa, in Olivier, E., and Suleman, H. (eds): *Proceedings of 14th International Symposium on Electronic Theses and Dissertations (ETD 2011)*, Cape Town, 13-15 September. Available http://dl.cs.uct.ac.za/conferences/etd2011/papers/etd2011_webley.pdf

Witten, I. H., Bainbridge, D., and Boddie, S. (2001) Power to the people: end-user building of digital library collections, in Fox, E. A., and Borgman, C. L. (eds): *Proceedings of the 1st ACM/IEEE-CS Joint Conference on Digital Libraries*, ACM, 94-103. doi:10.1145/379437.379458