DSpace as an Open Archival Information System: Current Status and Future Directions

[1]Hewlett-Packard Laboratories, 1 Cambridge Center suite 12, Cambridge, MA 02142, USA
{robert.tansley, mick.bass}@hp.com
[2]Massachusetts Institute of Technology Libraries, Cambridge, MA 02139, USA
kenzie@mit.edu

Abstract. As more and more output from research institutions is born digital, a means for capturing and preserving the results of this investment is required. To begin to understand and address the problems surrounding this task, Hewlett-Packard Laboratories collaborated with MIT Libraries over two years to develop DSpace, an open source institutional repository software system. This paper describes DSpace in the context of the Open Archival Information System (OAIS) reference model. Particular attention is given to the preservation aspects of DSpace, and the current status of the DSpace system with respect to addressing these aspects. The reasons for various design decisions and trade-offs that were necessary to develop the system in a timely manner are given, and directions for future development are explored. While DSpace is not yet a complete solution to the problem of preserving digital research output, it is a production-capable system, represents a significant step forward, and is an excellent platform for future research and development.

1 Introduction

Increasingly, research and educational institutions produce born-digital output. As well as traditional document-based material, researchers and teachers produce data in more complex outputs, such as audio, video, legacy application data, statistical databases, software, and others. Presently, constituents of these institutions do not have a suitable place to put these digital objects such that they will be preserved for the long-term. Organisational changes and the now well-understood problems of digital preservation mean that much of this valuable output, and hence initial investment, is lost.

To begin addressing this need, Hewlett-Packard Laboratories and MIT Libraries collaborated on a two-year project to develop DSpace, an open source repository system for the capture and preservation of digital materials. DSpace was designed as a production quality system offering the breadth of functionality required for a long-term institutional repository in a relatively simple fashion. After several months of working with early adopter communities at MIT, DSpace went into production at MIT Libraries in November 2002, and has been available as open source software since that time.

T. Koch and I.T. Sølvberg (Eds.): ECDL 2003, LNCS 2769, pp. 446–460, 2003.

Of course, we cannot claim to have solved the problem. Of those issues that are technical, many are at present research problems. DSpace is serving as the starting point for a number of development and research efforts looking into these issues.

The focus of this paper is to document the present functionality of DSpace, using the Consultative Committee for Space Data Systems' Reference Model for an Open Archival Information System (OAIS) [5] as a basis to demonstrate how far DSpace is towards becoming a comprehensive long-term institutional repository system. Decisions and necessary tradeoffs made during the development to meet production deadlines are explained. Also described are some thoughts on possible future research and development on the DSpace platform.

2 Related Work

DSpace draws on previous work and experiences from the field of digital libraries and repositories. The DSpace architecture has roots in Kahn and Wilensky's Framework for Distributed Digital Object Services [4], as well as Arms et al.'s work on digital library architecture [1], [2].

Comparable systems to DSpace are surprisingly few. The EPrints system developed at the University of Southampton [12] has many similar features to DSpace, but is targeted at author self-archiving of document-style material rather than long-term preservation of digital material. Interoperability with EPrints is of course desirable, and can currently be achieved through use of the OAI metadata harvesting protocol [11]. The Greenstone software from New Zealand Digital Library Project at the University of Waikato [13] is another open source digital library tool with a focus on publishing digital collections.

3 The Open Archival Information System Reference Model

One very relevant piece of prior work is CCSDS' Open Archival Information System (OAIS) reference model [5]. This describes the functional components of a system intended to preserve information for the benefit of a 'designated community'. It is a useful guide to the various aspects of archival systems, and the terminology is useful for describing existing systems, even if they do not exactly map onto the model.

The remainder of this paper describes DSpace's current capabilities in terms of these functional components. Note that DSpace's architecture does not directly correspond to the OAIS model. For example, archival information and data management information are both stored in the database. However, describing the functionality in terms of OAIS is a good way of expressing the functionality of the system to those familiar with the model. Additionally, since the OAIS represents mature thinking about what is really required to archive digital information, mapping DSpace functionality onto OAIS is a very effective way of finding out DSpace's strengths and where future work should be most urgently directed.

4 Archival Storage

Currently in DSpace, archival information is stored in a typical relational database, and the content itself is stored in the file system on the server. Thus, in DSpace, an OAIS Archival Information Package (AIP) is currently a logical object that is located in variety of database tables and files. This means that accessing archival information in DSpace is easy and efficient, but this is not optimal for preservation, since the representation information for much of the data is implicit in the DSpace code, and extracting an AIP requires the use of the DSpace code. Since we are using contemporary, actively maintained software and hardware, we feel this is an acceptable risk for the short term. To address this for the long term we are designing a means of storing DSpace AIPs in open, standard formats, for example based on METS [7] or RDF [8]. These will allow the reconstruction of the data in a DSpace given the AIPs, even if the DSpace code and runtime environment is not available. These would become the real archival storage in the system, and the information in the relational database would become access information. Since the logical contents are the same, the discussion that follows equally applies to both the relational database storage in place now and the future AIPs.

OAIS describes a specialisation of AIP called an Archival Information Unit (AIU) that represents the 'atoms' of information the archive is storing. In DSpace, the basic unit of archival storage, constituting an AIU in the OAIS model, is the **Item**. An Item is a logical work and can consist of many **Bitstreams**, the digital files that make up the content of the work. The structure of Items is depicted in figure 1.

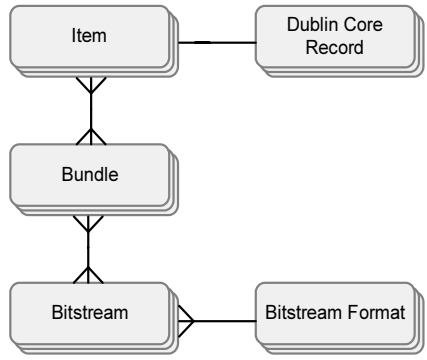

Fig. 1. DSpace Item Structure

Items all have one Dublin Core descriptive metadata record. This allows basic metadata interoperability across all of the items in DSpace. DSpace as shipped uses a derivation of the Library Application Profile, though the registry may be edited via means of an administrative user interface to suit an institution's particular need. This Dublin Core record also contains free text provenance information about who submitted the Item, and with which tool(s).

While Dublin Core is not nearly as descriptive as many of the metadata schemas and standards used by various communities, it is nearly always trivial to crosswalk from the richer description to Dublin Core. Serialisations of the richer metadata may be stored as Bitstreams within the item. Various efforts are under way to enable DSpace to make use of the richer metadata, such as the SIMILE project [9], which is investigating applying some or all of the Semantic Web tool stack and RDF.

Content within Items is divided into **Bundles**, each of which contains one or more Bitstreams. A Bundle may contain Bitstreams that are somehow closely tied, for example an HTML document containing images; the HTML source file and associated JPEG image files would be stored as separate Bitstreams within a single Bundle.

Presently, we do not have a way to express the exact relationship between such Bitstreams, for example expressing that a sequence of TIFF images are ordered pages of a book. Expressing and storing this part of the representation information for an item is a non-trivial problem, though efforts such as METS are well on the way to addressing it. We decided delaying the deployment and distribution of DSpace until a solution is implemented was unnecessary. We felt this was a reasonable, managed risk, since most of the initial content targeted for DSpace was in single document files or well-understood multi-file formats like HTML.

Bitstreams are individual digital content files, such as a JPEG image, HTML source document, PDF file, MP3 audio file, or MPEG movie file. Each Bitstream is associated with exactly one **Bitstream Format** from the system's Bitstream Format Registry. For each Bitstream some technical metadata is also stored: The Bitstream's size and MD5 checksum for integrity checking ('fixity information' using OAIS terminology), and an optional free text description of the file (always a useful fallback for catching any information that cannot easily be stored in a machine-processible way.) Currently, each Bitstream Format has the following information:

- A short name and a full description of the format, e.g. 'HTML 4' and 'Hypertext Markup Language version 4.'
- The MIME type, for example 'image/gif'. Note that many Bitstream Formats may have the same MIME type. For example, the MIME type 'application/msword' may be used by many different Bitstream Formats representing the output of different versions of Microsoft Word.
- Information for attempting to automatically recognise Bitstreams in this format. Currently this is consists of simple file extensions, for example '.jpg'. More sophisticated and reliable techniques would obviously be advantageous, so this piece of functionality is cleanly separated in the DSpace code.
- The 'support level' for the format. This is decided by the institution and indicates how well the hosting institution is likely to be able to preserve content in the format in the future, through migration or emulation. Factors influencing the support level for a particular format may be availability of detailed representation information (format specifications), availability of tools and source code for manipulating the format, available resources and demand from the institution's communities.

There are three possible support levels that Bitstream formats may be assigned by the hosting institution. The host institution should determine the exact meaning of

each support level, after careful consideration of costs and requirements. MIT Libraries' interpretation is shown in Table 1.

Note that the Bitstream Format registry does not contain representation information; however, to ensure preservation (a 'supported' format), the hosting institution should have extensive representation information for a format available and securely archived.

4.1 Archival Information Collections

The OAIS describes the concept of the Archival Information Collection (AIC), which is a kind of AIP that represents a collection of other AIPs. In DSpace, there are currently two kinds of AIC: **Community** and **Collection**.

Table 1. Bitstream Format Support Levels

Supported	The format is recognised, and the hosting institution is confident it can make Bitstreams of this format useable in the future, using whatever combination of techniques (such as migration, emulation, etc.) is appropriate given the context of need. In other words, the hosting institution has full representation information for this format and the resources to use it.
Known	The format is recognised, and the hosting institution will promise to preserve the Bitstream as-is, and allow it to be retrieved. The hosting institution will attempt to obtain enough information to enable the format to be upgraded to the 'supported' level.
Unsupported	The format is unrecognised, but the hosting institution will undertake to preserve the Bitstream as-is and allow it to be retrieved.

A DSpace Collection consists of DSpace Items that are somehow related. A Collection might be a technical report series, or a series of statistical data sets produced during a research project.

DSpace Collections are organised into Communities. These typically correspond to a laboratory, research centre or department. This structure is depicted in figure 2. Note that Collections may appear in more than one Community, and Items may appear in more than one Collection.

Fig. 2. DSpace Communities, Collections and Items

In DSpace, the each Community and Collection has some metadata corresponding to the OAIS Collection Description and Preservation Description Information. This includes the name of the Community or Collection, a free-text description of the content and/or purpose of the collection, and a free-text provenance description.

4.2 Identifiers

One important aspect of preservation is naming; archiving something isn't particularly useful unless it can be referred to and accessed. The OAIS describes two sorts of names: AIP Identifiers and Content Information Identifiers. Presently, DSpace uses CNRI's Handle system [10] to identify Items, Collections and Communities. Since these objects are stored in the relational database and are both archival and access-optimised, these Handles in are effect both AIP Identifiers and Content Information Identifiers. When AIPs are stored in an open, standard format outside of the relational database, the Handles are likely to become the Content Information Identifiers, and the AIP Identifiers may become something different. It may be that for each Item there exists a number of AIPs, corresponding to different storage formats and versions of that Item; the exact details have not yet been nailed down.

The advantages of using Handles are guaranteed global uniqueness, and a global resolution mechanism. Even if for some reason the global resolution mechanism becomes unavailable, the fact that the Handles are globally unique is still valuable, and they may still be used within DSpace to locate items.

DSpace does not assign Handles to subparts of Items, such as Bitstreams. While direct, programmatic access to subparts is often desirable, providing access in a preservable way is very difficult, since exactly what the intended subpart is may be unclear. Is it the particular bit-encoding or a higher-level concept such as 'the image on page 3,' in which case a bit-encoding in an appropriate contemporary format is required? Thus we decided it was inappropriate to give Handles to entities while we have no way of expressing exactly what it is that is being identified by the Handle.

Bitstreams within an Item do have a numerical ID (a database primary key) that is unique to a DSpace instance. Thus, as a compromise, individual Bitstreams may still be accessed directly using the numerical ID if necessary for a particular purpose. Modifying the code to allow Handles to be assigned to Bitstreams would be a trivial change, if such a path was deemed appropriate in the future.

4.3 History Information

While the provenance information in the form of prose in the DC record is very useful, it is not easily programmatically manipulated. DSpace includes a History system that captures a time-based record of significant changes in DSpace, in a manner suitable for later repurposing, for example to create audit trails and track where some change to information in the archive might have been changed.

Currently, the History subsystem is explicitly invoked when significant events occur (e.g., DSpace accepts an item into the archive). The History subsystem then creates RDF data describing the current state of the object. The RDF data is modelled using the ABC Model [4], an ontology for describing temporal-based data.

4.4 A Sample DSpace AIP

Figure 3 shows an example DSpace Item as an AIP. Preservation Description Information is shown in italics. Representation Information for the Bitstreams is not held within the system, but is linked by reference to the Bitstream Formats.

Representation Information for the AIP as a whole is not stored explicitly, as has been previously discussed. When AIPs are generated and stored using the METS, this will be addressed.

4.5 Migration and Disaster Recovery

Important aspects of the archival storage part of an OAIS system are backup and disaster recovery. These are not directly addressed by the DSpace code; the wealth of existing backup and recovery tools available for different operating systems may be employed with DSpace without difficulty. At MIT Libraries, the server's file systems are backed up to magnetic tapes which are then shipped off-site. A drawback with this approach is that disaster recovery would rely on the availability of the appropriate hardware and software, since the data would be recovered in the PostgreSQL database storage formats. This is an acceptable risk while we are using actively maintained hardware and software, however backing up DSpace AIPs in an open, standard format such as METS as discussed above is necessary for reliable long-term recovery.

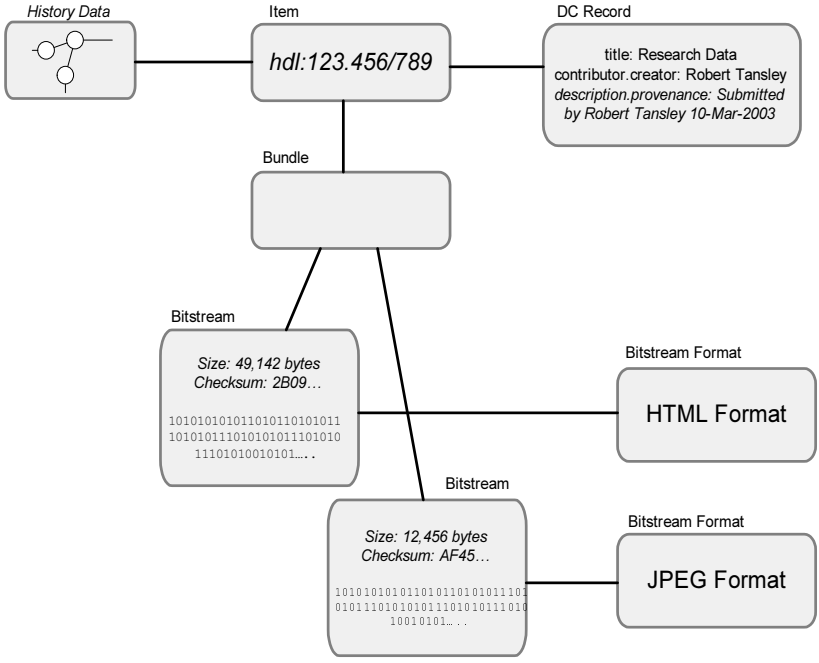

Fig. 3. A Sample DSpace Archival Information Package

Storing DSpace AIPs in the standard format would also make migration between physical storage media simpler, since it would not be necessary to use exactly the same database application with the new storage media.

5 Data Management

DSpace has a number of functions that fall within the data management part of the OAIS functional model. These include records of users of the system, authorisation control, and metadata indices.

5.1 E-people

DSpace holds information about users of the system for authorisation and authentication, for keeping records of who did what, and for subscriptions (described in section 8.2.) In OAIS these are termed 'Customer Profiles'. DSpace calls them **E-people**. These E-person records are not 'authority records' for authors and contributors to archived content. The term 'E-person' is used as opposed to 'user' since many parties accessing DSpace may be machines rather than real users.

E-people may also be organised into **E-person Groups**, so that authorisation policies and so on can be easily administered.

5.2 Authorisation

DSpace features a fine-grained authorisation system that enables individual E-people and E-people Groups to be permissions on individual objects in the system. In order for a user to perform an action on an object, they must have permission; DSpace operates a 'default deny' policy. Different permissions can apply to individual Bitstreams within an Item. Permissions do not 'commute'; for example, if an E-person has READ permission on an Item, they might not necessarily have READ permission on the Bundles and Bitstreams in that item.

5.3 Authentication

In general, the access mechanisms that people use to interact with DSpace are responsible for authentication. For example, when a user interacts with DSpace via the Web user interface, it is the Web user interface code that is responsible for determining who the user is. This is because authentication mechanisms tend to be specific to interfaces. MIT Libraries use X509 certificates to authenticate users; this mechanism may not be applicable to a Web services interface.

Naturally, the Web user interface code, and any other code that must authenticate a user, can and will make use of the E-person records held within the database.

The Web user interface code shipped with DSpace has been designed to make interoperating DSpace with any local authentication schemes as simple as possible.

5.4 Indexing

DSpace stores storing objects and metadata in normalised relational database tables, allowing efficient programmatic access and retrieval. In addition, DSpace maintains two kinds of index for access services.

The first kind is an ordered index of Items (by title or date) or authors of Items in the archive. This allows access mechanisms such as the Web user interface to retrieve

and present to the user ordered lists of objects to browse through. These indices are maintained in the database.

The second kind of index is a quick look-up index for responding to user's keyword search queries. DSpace uses the open source Jakarta Lucene search engine to manage and query this index. Presently, a subset of the Dublin Core fields is indexed. Lucene supports full-text indexing but DSpace does not use that feature yet.

6 Administration

Presently, DSpace administration is performed centrally by staff knowledgeable of the system and how it works. Future development will allow some of the administrative functions of the archive to be performed by individual communities if desired. This allows communities to retain a sense of control over their content in the archive, and to relieve central administrative burden.

Examples of the sort of administration that communities will be able to perform are authorising submitters, creating Collections, assigning roles in the submission workflow such as reviewers, and changing cosmetic aspects of the Web user interface specific to their Collections.

It was decided to have all administration performed centrally for now since the number of communities at MIT using DSpace is currently quite small, and it would have taken too long to develop a user interface sufficiently user-friendly and robust. Such an interface would have to ensure that mistakes made by community administrators do not cause damage.

There are two main means of administering the DSpace archive. Most administration can be performed via an administration Web UI. Some low-level tasks are performed by running shell scripts and commands on the server machine.

The administration Web UI contains tools for:

- Creating and editing metadata for Communities and Collections
- Creating and maintaining E-person records
- Controlling authorisation Groups and access control lists
- An interface for modifying Items in the archive. Although this allows any edits to be made to the item, in general it is just used to correct spelling mistakes and the like
- Withdrawing Items from the archive.
- Viewing and editing the registry of Dublin Core elements and qualifiers. Presently, this must be done with care, since various parts of the system such as the submission UI expect particular elements and qualifiers to be present
- View and edit the Bitstream Format registry
- View and abort active submission workflows

Tasks that must be performed via shell scripts and commands include:

- Importing many SIPs at once (batch importing)
- Administering backups
- Search and browse index maintenance

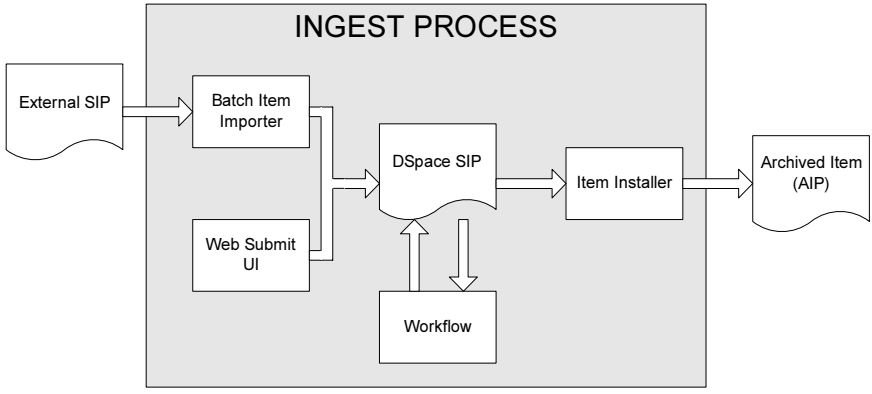

Fig. 4. The DSpace Ingest Process

7 Ingest

The basic task of the ingest part of an OAIS is to receive a Submission Information Package (SIP) and transform it into an AIP suitable for inclusion in archival storage. The DSpace ingest process is depicted in figure 4. At present, DSpace supports two methods of ingest.

Firstly there is a batch item importer. For this importer we have defined a simple external SIP format, in which the metadata (Dublin Core) is stored in an XML file, in a directory with the associated content files. The batch importer takes these external SIPs, and converts them into DSpace SIPs stored in the DSpace database and Bitstream store. Currently, content to be imported must be copied to a temporary directory on the server running DSpace, and the batch importer tool run via a remote shell.

Secondly, DSpace features a Web-based submission user interface. Using this, a submitter (producer using OAIS terminology) fills out appropriate metadata fields, and uploads the content files to DSpace. This information is used by the submission UI to construct a DSpace SIP in the database and Bitstream store.

Once the SIP has been constructed in DSpace, the workflow associated with the target collection is started. This corresponds to the 'quality assurance' function in the OAIS model. Presently, the workflow system in DSpace is quite simple. There are three possible workflow roles:

1. *Reviewers* can accept the submission, or reject it
2. *Approvers* can accept or reject the submission, and can edit the metadata of the item
3. *Editors* can edit the metadata of the submission, but may not reject the submission; when they have performed any necessary edits, they commit it to the archive.

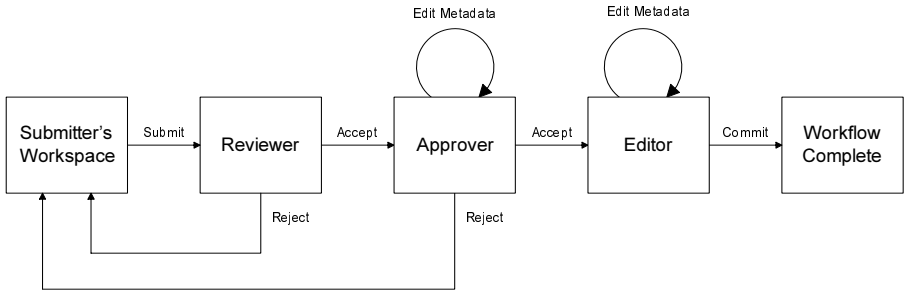

Fig. 5. DSpace Workflow Process

An incoming submission passes through these three steps in order, as depicted in figure 5. Zero or more E-people can be assigned to each role for a collection. If more than one E-person is assigned to a role, only one E-person actually has to perform it. In this case, the E-person performing the role 'locks' the submission so that the other E-people in that role know it is being worked on. If no E-people are assigned to a role for a collection, the submission passes straight through that step.

This relatively simple workflow will not cover every conceivable submission workflow, but does cover the needs of most communities. The exact semantics of each role may be determined individually for each Collection. For example, the Reviewer of one Collection might simply check that the uploaded files have been received correctly. The Reviewer of another might look at and make recommendations about the content of the submission.

Once the workflow process is successfully complete, the Item Installer is responsible for transforming the DSpace SIP into a DSpace AIP. This performs the 'Generate AIP,' 'Generate Descriptive Info,' and 'Coordinate Updates' functions in the OAIS model. Specifically, the operations it performs are:

- Assigns an accession date
- Adds a "date.available" value to the Dublin Core metadata record of the Item
- Adds an issue date if none already present
- Adds a provenance message (including bitstream checksums)
- Assigns a Handle persistent identifier
- Adds the Item to the target Collection, and adds appropriate authorisation policies
- Adds the new item to the search and browse indices

DSpace contains Java APIs for constructing a DSpace SIP, and invoking the workflow system and Item Installer. Thus, an institution may implement their own submission mechanism by building on those APIs. For example, a department within an institution might have an existing application for submitting content to a departmental Web site. A bridging application might automatically take new content from the departmental Web site and use the DSpace APIs to add the content to DSpace.

Another possibility is to convert metadata and content exported from another system into the simple DSpace XML SIP format for the batch import tool.

8 Access

There is little point in archiving things if they cannot later be accessed! DSpace offers a variety of access and discovery mechanisms.

8.1 Dissemination Information Packages (DIPs)

DSpace presently can disseminate three sorts of DIP: Community, Collection and Item record Web pages, individual Bitstreams via HTTP, and a file system based format including the metadata in an XML file and composite Bitstreams.

8.1.1 Community, Collection, and Item Record Web Pages

Each Community and Collection in DSpace has a 'home page' in the Web UI. This displays information such as the name and description, most recent new Items, and access to common functions such as submission and subscriptions.

For each Item in DSpace, the Web UI can present an Item record display page. This displays a subset of the Dublin Core metadata associated with the Item, a link to a version of the page displaying all of the Dublin Core metadata, and a list of Bitstreams within the Item which can be individually downloaded and viewed. Essentially, this Item record display page is the default DIP of each Item in DSpace.

If an Item has been withdrawn for some reason, the default DIP is a 'tombstone' page, which explains that the Item has been withdrawn and is no longer accessible.

8.1.2 Bitstreams

Besides the Item record display page, DIPs available to users take the form of simple, individual Bitstream downloads via HTTP or HTTPS. The MIME type of each format being downloaded is sent in the HTTP header, allowing the browser to select the most appropriate view or application for displaying the Bitstream to the end user.

DSpace currently has no more sophisticated DIP mechanisms than this. There is no tool that negotiates an appropriate Bitstream Format to send to the user or that can transform between Bitstream Formats. For now the communities using DSpace tend to use the same contemporary formats, so this is only a barrier to a small minority of users. The need for transformations is likely to increase over time as DSpace is used to archive more complex rich media material, so this is obviously an important development area for DSpace. A possible solution is a mechanism whereby available transformations are determined by the system, the user selects one (or one is somehow negotiated automatically), and the system then either performs the transformation and disseminates the results or sends a cached copy.

An additional possibility is that Bitstream(s) are embedded in some sort of emulation environment that is disseminated to the end user. For example, a software executable might be sent embedded in a Java Applet that emulates the software and hardware environment that executable runs in.

8.1.3 Batch Item Exporter

DSpace also features a batch export tool which can write DIPs for desired Items in the same XML format and directory structure that the batch import tool recognises. This

allows 'round trips' Items between DSpace instances. The exporter must be run using a shell on the server computer, and thus is not in general available to end users.

While DSpace does not have a standard, open AIP storage format, this is also a way of backing up data in the archive so that the DSpace software and hardware does not have to be available to reconstruct the archive in the event of some disaster. The forthcoming open, standard AIP format mentioned in section 4 above may be used as a DIP, probably replacing the current, simple format.

8.2 Discovery Mechanisms

DSpace already provides a variety of discovery mechanisms:

Handle Resolution. The CNRI Handles assigned to new DSpace Items may be resolved using the global Handle resolution mechanism. A Handle resolution request will result in the URL of the relevant Item record display page. In the future, Handle resolution requests may result in a variety of URIs or other means of accessing different available disseminations of a particular Item, Collection or Community.

Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). As more institutions bring up DSpace and similar institutional repositories, the resultant availability of on-line material presents a great opportunity for cross-institution access and discovery. One important effort in realising this is the Open Archives Initiative (OAI) [11]. DSpace functions as an OAI data provider; it will respond to OAI-PMH requests with appropriate descriptive metadata (Dublin Core.)

Search and Browse. The DSpace Web UI provides an interface to the search index. End users can enter search terms and view matching Items. These searches can be scoped to particular Communities and Collections. The Web UI also allows users to browse through the lists of Items ordered by author, title and date.

Subscriptions. Any E-person registered with DSpace can elect to 'subscribe' to one or more collections. When new Items appear in any of their subscribed Collections, the E-person is sent an e-mail summarising these new Items.

OpenURLs. DSpace supports the OpenURL protocol in a rather simple fashion. DSpace can display an OpenURL link on every Item record page, automatically using the Dublin Core metadata. Additionally, DSpace can respond to incoming OpenURLs. Presently it simply passes the information in the OpenURL to the search subsystem. The relevant item (if it is in DSpace) is usually at the top of the list of results.

9 Transformation

A final aspect of digital preservation that is not specific to any particular part of the OAIS functional model is the necessary migration of content in one format to another, as the hardware and software used to access digital material changes over time. In OAIS terminology this is referred to as Transformation.

At present, DSpace as shipped does not have any facility for automated Transformations. Although this seems like a serious omission, it is not a barrier to operating a DSpace; a suitable update will be available before any hardware, software and Bitstream Format specifications become truly obsolete and unknown.

The future Transformation functionality is likely to take the form of some administrative tool which is given some heuristic to decide what needs to be transformed. This may be as simple as a particular Bitstream Format, or a pattern of Bitstream Formats within a Bundle. The tool to perform the actual transformation would also be specified.

The Transformation tool would then apply the transformation to each Bitstream or set of Bitstreams in the appropriate archive, Community or Collection. These new Bitstreams would be stored in new Bundles within the Item, and hence available through the same Handle as the original(s).

Transformations are often 'irreversible' and may result in the loss of some information. It is not always possible to represent every piece of information that can be held in one format in another. Hence, the following precautions would be taken:

- Bitstreams that are superseded by the Transformation would not be removed, so that any future, improved version of the Transformation can be applied. The superseded Bitstreams might be moved to cheap 'off-line' media, however.
- The provenance information (currently stored in the Dublin Core record) of the Item would be updated to include information about the Transformation.
- The History system would store information about the Transformation event, including when it occurred, the tool used, the person who initiated it, and the state of the Item and its contents before and after the Transformation.

Should the Transformation later prove to have been problematic, for example if it resulted in loss or garbling of important information, the data stored by the History system could be used to automatically 'back-track' the Transformation, and re-apply an improved Transformation tool. Since the History system stores the state of the Item before and after Transformation, it is also possible to verify that Transformations have been successfully undone.

10 Conclusion

With DSpace, we feel we have made significant progress towards providing a tool to address an institution's long-term digital repository needs. It is not yet a complete solution, and by considering DSpace in the context of the OAIS model, we have been able to highlight the areas in which DSpace needs work:

- Describing complex relationships between Bitstreams (Bundle descriptions)
- It would be relatively easy to write a tool that migrated Bitstreams from one format to the other using the DSpace Java API. Preferable would be a standardised way in which this can be achieved so that migrations and transformations can be shared between DSpaces, and so the effort required to perform future migrations is small.
- We need a way to use these complex relationships between Bitstreams and transformation tools to provide more useful disseminations, especially as the complexity of data and media in the system increases. The FEDORA work [6] in particular provides a useful framework for addressing this issue.

By working with a host institution we have been able to focus on the providing the functionality required to run as a live service now and to immediately start capturing digital information that would otherwise be lost. In building DSpace as a simple, modular system, we have given future research and development in the above areas a

head start, since it is clear where in the system each piece of functionality should reside. Additionally, since DSpace is open source, it can potentially provide value to a great many institutions, and stands to benefit from a large community of developers and information science professionals.

We are confident that DSpace represents a great step forward in the fight by institutions to capture the digital output of research investment, and is in an excellent position to provide a comprehensive solution to this problem in the future.

References

1. William Y. Arms: Key Concepts in the Architecture of the Digital Library, *D-Lib Magazine*, July 1995 <http://www.dlib.org/dlib/July95/07arms.html>.
2. William Y. Arms, Christophe Blanchi, and Edward A. Overly: An Architecture for Information in Digital Libraries, *D-Lib Magazine*, February 1997 <http://www.dlib.org/dlib/february97/cnri/02arms1.html>.
3. The GNU EPrints Software <http://software.eprints.org/>
4. Robert Kahn and Robert Wilensky, *A Framework for Distributed Digital Object Services*, May 1995 <http://www.cnri.reston.va.us/home/cstr/arch/k-w.html>.
5. Consultative Committee for Space Data Systems, *Reference Model for an Open Archival Information System (OAIS)*, CCSDS 650.0-R-2, Red Book, Issue 2, July 2001 <http://ccsds.org/documents/pdf/CCSDS-650.0-R-2.pdf>.
6. Sandra Payette and Carl Lagoze, "Flexible and Extensible Digital Object and Repository Architecture," in *Research and Advanced Technologies for Digital Libraries: Proceedings of the Second European Conference, ECDL '98*, Crete, Greece, 1998., G. Goos, J. Hartmanis, and J. van Leeuwen, eds., *Lecture Notes in Computer Science*, 1513 (Berlin: Springer, 1998) <http://www.cs.cornell.edu/payette/papers/ECDL98/FEDORA.html>.
7. Metadata Encoding and Transmission Standard (METS) <http://www.loc.gov/standards/mets/>
8. W3C Resource Description Framework (RDF) <http://www.w3.org/RDF/>
9. SIMILE: Semantic Interoperability of Metadata and Information in unLike Environments <http://web.mit.edu/simile>
10. Handle System Overview. <http://www.ietf.org/internet drafts/draft-sun-handle-system-10.txt>
11. The Open Archives Initiative <http://www.openarchives.org/>
12. The Greenstone Digital Library Software <http://www.greenstone.org/>