FEATURES AND FUNCTIONALITIES

Once you have a general idea of how the IR will be used and made some decisions regarding the type and extent of services you intend to offer, you are ready to start defining the features and functionality of your ideal IR.

This chapter walks you through some larger feature and functionality decisions, discussing not just the technological aspects, but also some of the accompanying policy decisions and implications. This exercise should help you generate a checklist of desired system features and functionalities that can serve as your guide when you are ready to select an IR system.

Open source or commercial

The term *open source* has been bantered about a great deal. For a system to be open source it should have several characteristics. The first is that the software is free and requires no royalties or fees to be used. A fee may be required for customer support, such as the services that Red Hat offers the Linux platform, but the software itself is free and usually downloadable from open-source distribution points, such as Sourceforge.

Red Hat, www.redhat.com

Sourceforge, http://sourceforge.net

An open-source system also has source code in a form that is easily obtainable. The source code provides for the third characteristic of an open-source system: that the software is modifiable and allows for derived works.

An open-source system is one for which the software is available without cost and is in a format that allows the user to make changes easily. For a more detailed explanation, see the Open Source Initiative's website.

Open Source Initiative, www.opensource.org/docs/definition.php

Although the majority of available IR systems are open source, carefully weigh the significant tradeoffs with open-source systems. That an open-source system is free is appealing. But just as gift books are not free to a library, so too is an open-source system not free to its user.

When using an open-source system, an organization must be self-reliant. No vendor or company is at the other end of a phone line to answer questions. Open-source systems also can lack detailed documentation and manuals.

Generally, the users of open source systems must rely on one another for technical support and assistance, often through the use of list serves and user groups. When considering an open-source product, the institution must ensure it has the staff and expertise to install the software, customize it as needed, and, most importantly, keep it up and running. These significant staffing costs could easily over time exceed the purchase price and yearly maintenance costs of a commercial system.

On the other hand, if an organization has some unique needs for an IR, then an out-of-box commercial system wouldn't likely meet those needs. For example, a university may want to integrate its institutional repository's ETD collection with an already existing system in the graduate school that handles the movement of dissertations through the official submission process.

In this case, an open-source repository system may be the best choice because the programmers would have access to the IR system's source code and be able to build a crosswalk for data between the two systems.

If you purchase a commercial IR system, consider the question of hosting. Particularly for small institutions with limited technical staff, having both the system and the document storage hosted by the vendor is desirable.

This combination relieves the institution from allocating any staff to the institutional repository, other than those required for planning and training. The downside to hosting is that the institution is placing its IR in the hands of an organization over which it has no control.

Questions to ponder:

- Can the IR project budget better sustain additional in house staffing or commercial vendor fees?
- Do you have the appropriate infrastructure to support an IR system, such as a data storage backup and user authentication systems?
- How unique is your environment and the predicted uses of the system?
- How much experience does your IT staff have working with open-source systems?
- What will you do if your commercial vendor goes out of business or is bought by another company that has different strategic interests?

Format types

The list of possible digital format types is ever-growing. Some IR systems are format-neutral—the system accepts deposits of any digital format. Other systems are hard-coded to only accept formats of certain types, and still others provide the system administrators with the ability to restrict acceptable format.

The breadth of acceptable format types must be balanced against the institutional commitment to preserving all of them. If your organization does not want to face the prospect of trying to migrate Quark or little-known executable files in five years' time, then a system that can prevent the deposit of these files into the IR is desirable.

Another feature to consider is automatic format conversion. On deposit, some systems can automatically convert some less preservation-friendly formats, such as Word, into PDF or HTML, which have a higher likelihood of long-term preservation. By storing the native Word file, along with the newly created PDF file, you now have two potential avenues for future migration.

Questions to ponder:

- Are there boundaries to the scope of your preservation commitment?
- Do you want to be able to control which format types are acceptable for submission?

Deposit structure

Another facet to consider is the potential complexity of the deposit structure. For some systems, a deposit consists of only a single file, but for others any number of files can constitute a deposit. An example of a common multifile deposit is an ETD, which may contain, for example, a PDF, accompanied by audio and video clips, a few images, and a large dataset.

Not all systems will allow all these files to be deposited together as one single submission. Or, if they can be deposited together, they may not be stored in such a way that they can continue to interact with one another.

For instance, although a system may be able to handle the deposit of the multiple files that make up a Web page, once deposited the file associations are lost, making the correct rendering of the Web page impossible.

Another feature to consider is whether an IR system can upload multiple files at once or if each file must be uploaded sequentially.

Questions to ponder:

- Will potential deposits consist of single or multiple files?
- If there will be multiple file deposits, will the files need to continue to interact after they are deposited?

Versioning

Versioning is a prime example of an IR system feature that is intricately related to your policy decisions. If your policies restrict the use of the IR to finished, rather than in progress, works, little if any need should exist for a versioning system.

Invariably, however, potential users of the IR will ask you if it can handle multiple versions. This concern is due in large part to the need that most people have for an authoring system, as discussed in Chapter 3.

If your policies permit multiple versions of a work in the IR, then you need to decide whether you want an IR system that stores and makes evident the relationship of multiple versions or whether one version can be replaced by a corrected version.

Questions to ponder:

- Is versioning an important feature or will the IR be for finished, completed documents only?
- Does a preprint accompanied later by a post-print version still subscribe to the guideline of "completed documents only?"
- If versioning is supported, should the persistent URL stay associated with the original version, the most recent version, or to the collection of versions?
- If your policies restrict the use of the IR to only finished works, does an IR with a versioning system send a contradictory message?

Access control

All IRs offer some mechanism by which access to the documents can be controlled. The robustness needed in access control depends on the types of potential documents.

ETDs are a great example of the possible degree of complexity. Once a dissertation is completed, the author may want the opus available to the entire world. Other authors, though, might want to turn their dissertations into a book, so they only permit their university community to have access. Still others will be

applying for a patent based on the findings of the dissertation, so they only allow their dissertation committee to see the finished work.

Or perhaps the dissertation includes copyrighted images. Although the author may have the right to include the copyrighted images in the dissertation as permitted by fair use, the author does not have the right to distribute those images freely to the world along with the rest of the dissertation via the IR.

To protect copyrighted material, the text of the dissertation may need to have a different level of access than some of the images included within it. ETDs provide an extreme example of the complex level of access permissions that may be required.

Where and how access control levels are determined is another feature to consider. For some IR systems, the system administrator assigns access at the collection level, forcing all documents within a collection to share the same access levels. Other systems place designation of access levels in the hands of the depositors at a file level.

Questions to ponder:

- What types of documents will be deposited into the IR and do they require varying access restrictions?
- Will all deposits be simple, single-file deposits or are complex, multifile deposits possible?
- Is a system in place to identify classes of community members or can the community of users be defined by IP?
- Will you need to assign access control at the repository, collection, deposit, or file level?

User interface

Carefully consider several aspects of an IR's user interface when selecting an IR system. First is ease of use. No matter how sophisticated and elegant the backend of an IR system may be, it is practically useless without a usable interface on the front end.

The total number of potential end users dictates how important ease of use needs to be—the larger the pool of potential users, the more important is ensuring that the interface is clear, simple, and easy to use.

When considering an IR system, inquiring about whether usability testing has been done on the user interface is not out of scope. (Keep in mind that with an open-source system, someone other than the original programmers of an open-source system may have conducted the usability testing, so be sure to post the question to the system's user group.)

Also, consider the metaphors that an IR system employs in its user interface. Design metaphors have evolved that help the end users through the initial learning curve. A common example is the folder metaphor for the organization and storage of digital files. An IR system that employs familiar metaphors in its user interface likely requires less hands-on training.

Another feature of the interface that you should consider is the level of branding and customization available. The interface of an IR should reflect the culture and image of the institution and not the culture and image of the original programmers.

This customization may mean minor changes, such as the use of the institution's official colors and logo, or it may mean a complete reworking of the interface so it blends seamlessly with the institution's official website. Systems that use cascading or extensible style sheets are much easier to customize than those that require a programmer to tinker with every page.

Although you may want your IR to have a consistent look, you need to balance this desire against the desires of your users to be able to personalize their parts of the IR, such as a political science department's working paper collection. This flexibility requires an IR system with templates that can retain the overall look of the interface, while still permitting some personalization.

Questions to ponder:

- To what extent should the IR reflect the institution's website?
- How important is branding the interface?
- Will you want to use templates to retain some control over the look of the IR, or can each collection have its own, unique user interface?
- Are there interface design metaphors with which your community is already comfortable?

Authentication or off-site access

Most large organizations have some form of an authentication system in place. Understanding the requirements of your organization's identity management system, such as LDAP or Shibboleth, can ensure that the IR system you select is compatible. If they are not compatible, in the worst case you will have to manually create and maintain an account for every potential user of the system.

Most organizations, particularly universities, have a remote access system to permit off-site users access to restricted materials, such as licensed library databases. If you want to ensure that your users can obtain and submit documents to the repository while off-site, then compatibility between your IR and your remote access system must exist.

Questions to ponder:

- Will you need to provide for remote access to the IR?
- How is authentication handled at your organization?
- Will there be any access restrictions to the collections? If so, how complex an array of access will you need to sustain (such as worldwide, institution-wide, department-wide, or geographically disbursed people)?

Editors and administrative levels

For some collections, you may want or need the ability to create collection editors, who can perform functions such as metadata control and ensuring quality of the content.

For example, each deposit to an image collection is placed in a virtual holding area until someone checks the image to ensure the preferred format and dpi were used. Another person verifies that including the image in the collection does not violate copyrights. A third person assigns appropriate metadata to it.

LDAP: Lightweight Directory Access Protocol

Shibboleth is developing architectures, policy structures, practical technologies, and an open-source implementation to support interinstitutional sharing of Web resources subject to access controls. (Source: http://shibboleth.internet2.edu)

dpi: dots per inch

Only after these three people virtually sign off on the image does it become an official document in the collection. A dissertation committee and graduate school also could use this editing feature to sign off on an electronic dissertation.

Similarly, a conference committee could use an IR to gather paper submissions—only allowing those submissions the selection committee approves to become part of the collection.

On the other hand, just the presence of this gatekeeping functionality can cause fear and anxiety about censorship and loss of intellectual freedom.

Faculty, in particular, are autonomous people within the academy. An IR system with even the potential of gatekeepers will cause some faculty members to reject the system on principle. Meanwhile other faculty members will welcome the functionality because it allows them to delegate the submission task to a student or administrative assistant, while still retaining ultimate quality control over the collection.

With the establishment of each new collection comes a series of administrative functions, such as designating authorized submitters. In a large organization, where the community members could potentially create hundreds of collections, distributed administrative roles are a necessity.

But if an IR system supports only one class of system administrators, you may not be able to push down any administrative functions without placing system-critical functionality into the hands of inappropriate people.

Questions to ponder:

- Will some person or group of people have submission approval rights?
- Will the IR be used for dissertations, conference proceedings, or other materials that may require gatekeepers?
- Will you be offering metadata enhancement or proxy submission services that would require submission-editing functions?
- Will one class of administrators be sufficient to support your IR?

Environment and infrastructure compatibility

Particularly when using an open-source IR system, ensuring that available staff are comfortable programming in the system's language is critical. If your staff is fluent in Perl, but the IR system is programmed in Java, the project's timeline may be significantly set back while your staff becomes comfortable with the system's code or a new staff member is hired.

Similarly, if the computer center is comfortable working with a Sun Solaris platform, then it would be to your benefit to find an IR system that can run on Solaris.

In addition to compatibility with your staff's skills and expertise, ensure compatibility of the system with your institution's infrastructure and computer environment. Determine whether compatibility with your existing firewalls, authentication, and redundancy systems will put requirements on the IR system. Ensuring the compatibility of the IR system with your current computing environment can save a great deal of time and money.

Questions to ponder:

- If you are planning on using existing technical staff members to run the IR, what are their expertise?
- If firewalls are in place, can the IR system work within the firewalls?
- If you want the IR to live within the storage area network (SAN) or redundancy system that you have just paid several thousands of dollars to establish, what requirements do they have?

Clients

Significant trade-offs exist between using an IR system that requires the use of a client versus a Web-based interface. A benefit of a Web interface is the ability to access the system from any location, providing a remote authentication system is in place.

Client: A software program that can communicate and transfer data from another computer, in this case the institutional repository server.

A Web interface eliminates the need to install a client on the computers of all potential submitters, which can be time-consuming and potentially an endless task if client updates are frequent.

Clients, however, usually provide services that are more robust. For instance, a client could integrate the IR service into the users' desktop so the submission process can be accomplished through common desktop applications, such as Microsoft Word.

By using a client, the IR's server could become a mapped drive on a person's computer facilitating the seamless movement of documents from desktop to IR. A system that has a Web-based, general-user interface, but a client-based administrative interface, is quite common.

Questions to ponder:

- How many people will be making submissions into the IR?
- How many collection-level and system-wide administrators will there be?
- Are desktops centrally maintained or are individuals or departments in charge of the care of their own computers?
- Can updates be remotely pushed out to all desktops?
- Do you have an authentication system that will provide for remote access to the IR?

Metadata and protocols

Behind all IR systems is the metadata schema used to describe and identify each unique item in each unique collection. Existing IR systems differ widely in the handling of metadata. Some have created their own metadata schemas, and others have used or adapted existing schemas, such as Dublin Core or MARC.

Dublin Core, http://dublincore.org

MARC, www.loc.gov/marc

In some systems, the metadata schema is hard-coded, and others support some degree of modification. Some IRs can provide for a different metadata schema with each collection, and others require the same metadata schema across the entire repository. The variety of your potential collections determines the degree of metadata flexibility you need.

If you want your IR to share its data with other systems, such as your library's catalog, you need to select an IR system with the same or compatible (that is, existing crosswalk) metadata schemas. Otherwise, the use of your repository's metadata may be restricted to just the IR or require your technical staff to create one or more metadata crosswalks.

Similarly, if you would like to broadly advertise the content of your IR, ensure your repository system uses well-established protocols. A federated search engine, such as Endeavor's EnCompass and ExLibris' MetaLib, could include the content of your repository in its searches if the IR system is compliant with the Z39.50 protocol.

EnCompass, http://encompass.endinfosys.com

MetaLib, www.exlibris-usa.com/metalib.htm

Z39.50, http://lcweb.loc.gov/z3950/gateway.html#about

As discussed in Chapter 2, metadata harvesters, including the University of Michigan's OAIster, can harvest your IR's metadata if the system is compliant with the OAI metadata harvesting protocol.

Questions to ponder:

- Do any of your potential IR collections have specific metadata requirements, such as a locally created schema?
- Will the metadata schemas need to differ between collections in your IR?
- For what purposes will you want to share or transfer the IR's metadata?
- Will the IR be searched as a stand-alone repository, or do you hope to incorporate it as a source in federated searches?

Batch importing or exporting

The ability to batch load data into your IR may be a deal breaker when trying to acquire large pre-existing collections because the efficiencies and long-term benefits of the IR will be overshadowed by the enormous, time-consuming effort of rekeying several hundred metadata records.

Similarly, a batch-exporting system can be valuable if data needs to be copied into other databases, such as exporting the metadata for dissertations from the IR to the library's catalog.

Creating a batch-loading or exporting script is not too difficult a process, so long as you have proper access to the data tables and their structure, which may not be the case with a commercial system.

Questions to ponder:

- From where might you be extracting data? Sending data?
- If batch loading or exporting is required and the IR system does not have the functionality, do you have the staff to create the necessary scripts?

Persistent linking

Ideally, once a document is deposited into the IR, the system assigns a URL that remains the same throughout the lifetime of the document. Persistent linking is an important component in the overall preservation function of the IR.

Without reliance on a linking methodology, such as CNRI Handles or PURLs, links will break when documents are moved to different storage networks. Persistent linking also is critical to citation-based scholarship. Unfortunately, sometimes the trade-off for persistent links are long or complex URLs.

Questions to ponder:

- Is there a persistent linking methodology with which you are most familiar?
- How important is that the URLs be easy to remember, recognized, or convey information about the relationship of the document to others in the IR?

CNRI Handles, www.handle.net

PURLs, http://purl.org

Search engine

An IR is only as good as the discovery mechanisms it employs. Unless the appropriate and relevant content can be easily located, the IR fails to be an effective distribution mechanism.

Search engine capabilities of IRs can vary tremendously in their functionality. Some limit searching and browsing to just a few metadata fields, and others provide searching and browsing capabilities across all of them.

Although full-text searching of the documents themselves can be a desirable feature, it can add considerable cost and complexity to the system. Consider other features of the search engine such as result display format, sorting, and relevancy ranking.

Questions to ponder:

- Will you rely on the system's native search function, or do you have a pre-existing search service that you will want to place on top of the IR?
- How many documents do you predict will be deposited into the IR, recognizing that the greater the quantity of content, the greater the level of sophistication needed in the IR's search engine?

Usage statistics

Accurate, detailed usage statistics are an essential assessment tool. The quantitative data in the usage statistics is valuable in demonstrating an IR's use or lack of it. Moreover, the statistics provide the technical support staff with evidence as to whether the system's infrastructure is sufficiently robust to handle the level of use.

The statistics also can be of great value to the owners of the content. For instance, the number of accesses of an e-print helps assess the impact of a researcher's contribution to the field. As a result, faculty members may want to cite the usage statistics in their tenure portfolio.

Usage statistics of datasets can help support the argument that a grant's federally funded dollars served a public good. High ETD use would be persuasive when pitching a derivative of someone's dissertation to a book publisher. High usage also may be the most persuasive argument for why authors should continue to deposit their documents into the IR.

If the use statistics can be integrated into the user interface, they can act as a sort of rating or recommendation system, such as CNN's "10 Most Popular

CNN, www.cnn.com

Stories" link on its homepage. Clear, regular reports of usage statistics are enormously useful and should not be treated as an unessential luxury.

Questions to ponder:

- Are systems already in use by your community that provide usage statistics, and, if so, how are those stats used?
- What types of quantitative assessment measures will be used to judge the success of the IR?

Extensibility

Some IRs are meant to be stand-alone products, and others can interact with third-party systems with varying degrees of ease to extend the capabilities of the system.

You may want to sit various software, both commercial and open source, on top of the IR to extend its functionality. These needs will determine how important the degree of IR system extensibility is to your organization.

Luna Insight,
www.lunaimaging.com/insight/index.html

For instance, Insight, the commercial software developed by Luna Imaging Inc., provides some sophisticated tools for working with digital visual material. While using your IR to ingest the digital images, users may prefer to use Insight to browse, search, and manipulate the materials.

A system's level of extensibility can be partially determined by the presence of service interfaces and application program interfaces (API), for example. The degree of modularity of the system is another indicator, as is the extent to which the data table structure can be manipulated or altered.

Questions to ponder:

- Will the IR act as a stand-alone system, or may it be used as a repository that delivers content to other systems?
- Are there administrative systems in place, such as a dissertation approval system, that you will want to integrate with the IR?

When working through the preceding list of IR features and functionalities, decide which are most important to you, distinguishing between the *desired* and *required*. Also, add any of those that were overlooked.

Next, create a features and functionality checklist that can serve as your guide as you search for the IR system that best fits your needs. Or write a narrative description of your ideal IR system, against which you can compare those that are available. By predetermining your needs and wants before shopping, you can be sure to remain focused on your core requirements.