Access: Discovery and Delivery

Introduction

Repository access is about providing end users with material that serves their needs. Users in access systems can be humans navigating a GUI environment, humans querying the repository using machines through backend tools (API queries), or machines scraping the repository (API calls, OAI-PMH). Repositories should consider to what extent they allow access of content to multiple types of users.

Access is comprised of two functions: Discovery and Delivery.

- **Discovery**: Methods allowing user to find the information object that answers the question they were asking. Discovery services can use a combination of metadata and extracted content (e.g. OCR'd documents, audio transcripts) to answer the user's query.
- **Delivery**: Serving up metadata and digital object(s) as requested by the user. Delivery methods can be browser rendering, downloadable files, metadata returns.

Contents

- Discovery
- Delivery
- Mediating access
- User experience
- Accessibility
- Usability studies

Discovery

Discovery refers to the discovery of content from both within and outside the repository. Is your content discoverable from the open web? How can your content be searched from within by the repository's search functionality?

**Discovery from the open web**

Discovery in this sense is closely tied to the best practices of metadata and interoperability. The exposure of particular metadata fields and their representation greatly affects the ability of users (and machines) on the open web to find your content.

- Consult with metadata services about metadata standards designed specifically for open web discovery, like Schema.org.
- Work with developers to ensure that the markup of your pages includes tags and semantic markup that will improve the search engine discoverability of your content.

A documented, dependable API will make it easier for machine users on the web to consume or search your content. These users may be search engines, but they could also be researchers who want perform text analysis or reuse your content in other ways. API consumers can also be bad actors, overloading your server or using your content for undesirable purposes.

- Work with your designers and developers to create API access that will best serve your users and their needs.
- Document API access.

**Discovery within your repository**

Discovery within your repository should meet your users' needs and expectations as well as possible. Two kinds of data are important for discovery - assigned metadata and extracted content. Full-text search tends to be an important use case for repositories but extracted content can also be used to search images, audio visual material, data files, and other types of content. It is important to carefully understand the types of discovery that users need and expect for your content. Implementing full-text search for large and complex objects is non-trivial.

- In addition to configuration related to extracted content, a number of other discovery options and configurations should be considered:
  - Fields that can be sortable
  - Facets based on metadata
  - Metadata that can be clicked on to prompt a new search
  - Discovery of relationships between objects (e.g. content in same collection) and within objects (e.g. pages within a book)
  - Browse functionality
  - Shareability of specific content based on URL construction
  - Ad hoc grouping of content into collections
  - Controlling levels of discoverability for access-controlled objects, including withdrawn objects (see mediating access below)
  - Language support, including language-based configurations for stemming, spell checking and synonym dictionaries

- Identify discovery functionalities needs for your users.
- Identify discovery methods for repository content.
- Evaluate the configurability of your solution for accommodating new discovery requirements as they arise.
Test discovery mechanisms for accessibility compliance.

Delivery

Delivery is the method by which content discovered by user is served up to them. In a repository, delivery of content is contingent on file format(s), access and use restrictions, browser type, digital object size. Web delivery presents challenges due to the variability of environments the user may use to interact with the repository and its content. In order to grapple with these issues, there are two approaches to delivering content discovered within a repository.

- **Browser rendering of content**: users will be able to see and interact with the digital content without needing to download a file to view on their local computer.
- **Downloading content**: users download content to use material in an environment outside of the repository. Burden of rendering the content is on user and their computer environment.

Often there may be multiple delivery options for objects in a repository, and there are issues that affect delivery methods:

**Format being delivered**
Certain formats are more easily rendered in a browser than others. Certain formats researchers may want to work with outside the repository environment.

- Identify formats being delivered.
- Determine preferred delivery option for formats in repository.

**Accessibility**
Repositories should deliver content in an accessible way. This may mean delivering transcripts alongside video that are time coded together or ensuring large objects that are delivered through download are clearly identified with recommended connection capabilities.

- Identify accessibility requirements for formats and their preferred delivery method.
- Identify what kinds of technical platforms are and are not supported (computer browser, mobile applications, etc.)

**Aggregate vs. single item delivery**
Decisions about delivering a single item or aggregate objects will affect your data model. A single item may be easier to render in a browser within a repository than an aggregate object (e.g. a .zip file, .xsl file).

- Determine whether content can or should be delivered as single item or aggregate.

**Delivery restrictions**
Digital content may have restrictions that impact delivery options. These may be:

- Time specific (e.g. embargo)
- Location specific (e.g. only certain countries)
- Person or role specific (e.g. creator only; Cornell community only)
- Content use or reuse allowances (e.g. embed streaming audio or video on another website; downloading entire collection; downloading only low-resolution derivatives)

- Identify content restrictions that may impact delivery options.
- Document the types of restrictions on content that affect delivery.

Mediating access

Often content in repository may have limitations on access. Limitations may be on how content is discovered, when users can get content delivered, or even what types of users may get any access to the content. Mediation is managed through policy and authentication.

**Policy**
Mediating access through policy means setting rules governing acceptance of content into the repository (e.g. only open access content; no embargoes allowed; all content is available to certain designated community but not the open public). Policies help limit the complexity of access mediation mechanisms repository infrastructure requires.

- Establish policies defining supported access conditions for content.

**Repository authentication**
Repository authentication can mediate access to content based on policy, user identity, metadata about content, and rules defining logic to...
provide appropriate machine-actionable access. Authentication could be passive (e.g. IP restriction) or active (e.g. user login).

- Identify user type(s).
- Identify discovery restriction(s) for content in repository (who can discover what how?)
- Identify delivery restriction(s) for content in repository (who can get what content how?)
- Determine what policies govern access mediation.
- Determine what level of authentication is needed for content.
- Document rules needed to provide appropriate types of discovery and delivery to user(s).

**Personnel cost to mediating access**

Mediated access may also add an administrative burden to properly identify content requiring mediation and delivering content through a potentially more manual process. Self-serve access may require a more mature repository infrastructure.

- Identify conditions requiring mediated access.
- Evaluate potential personnel cost necessary to provide timely mediated access given demand of material.

**User experience**

The user experience of your repository is, by its nature, informed by many of the other principles described in this document. In particular, the user stories and features described during the initial definition of your repository and the metadata that powers your user experience are essential to providing discovery and delivery functionality to your users.

Successful discovery and delivery requires that you undertake the best practices of metadata outlined in elsewhere in this document and combine them with best practices from user experience research and design.

User experience practices to undertake include:

- Define your user community and then collaborate with user experience, research, and usability groups within CUL to develop personas, use cases and other UX deliverables to communicate your understanding of users and their needs to repository designers, developers, and maintainers.
- Conduct accessibility reviews of repository solutions and necessary remediation throughout the repository's life in service.
- Conduct usability testing and accessibility reviews to verify that repository design and development decisions meet your users' needs throughout the repository's life.
- Review usage statistics regularly to answer questions about the ways your audience is using the repository.
- Conform to best practices of interface design and maintenance. An appropriate, modern look and feel with clear institutional associations promotes user confidence and greater use of the repository. Repository solutions should always include the capacity to display appropriate Cornell University Library branding.

**Accessibility**

In order to provide equitable access, accessibility should be a primary consideration in repository access. Accessibility considerations include:

- Vision impairment
- Auditory impairment
- Dexterity impairment
- Language of user
- Cognitive abilities of user
- Mobile technology
- Internet speed

Cornell University Library works toward meeting a minimum of [WCAG 2.0 AA Standards](https://www.w3.org/WAI/standards/webcontent/). Any new web sites or repositories should meet this standard. The CUL User Experience team can advise you on accessibility best practices and connect you to other Cornell University resources. Facilities for testing access for disabled users are available on campus through student disability services. In addition, they offer regular trainings for web developers and others providing online resources to disabled students. Cornell IT now also offers a University-wide Site Improve license which provides automated accessibility testing.

- Identify accessibility minimums for repository based on user communities.
- Talk to CUL User Experience Team.
Be sure that your development team makes accessibility checks part of the development process and not an after thought. Many tools exist to help them with this task. Tools/checklists to consider: A11Y CLI Audit Tool, WCAG 2.0 QuickRef, the A11Y Project’s checklist, WAVE Browser Extensions.

Use university supported captioning for A/V collections: https://it.cornell.edu/vod/captions-recommended-method

Implement accessibility testing criteria.

Usability studies

Usability studies help evaluate whether users are able to perform the tasks required to interact with content in your repository. These studies provide a process to examine ease of use, intuitiveness, learnability and overall satisfaction that users experience when interacting with the repository. Repository systems are more complex than many other library websites because users may be both creating and consuming content, as well as sharing content, updating content and mediating the access of others to the content. All of these types of interactions should undergo usability testing. You may also have staff spending many hours working on the repository, testing for efficiency in their UX may be important to you as well.

CUL and Cornell have support in place for conducting user and usability studies. A good first step is to contact CUL’s Usability Working Group and CUL’s Research and Assessment department. These groups can help you define testing goals and implementations.

Initial development

The studies you conduct will be determined in part by your current development phase. For instance, early usability studies should involve low fidelity prototypes that demonstrate the in-progress nature of the repository's development, granting users the freedom to offer wide ranging feedback without fearing they will derail or insult your project. Later studies could include ethnographic observation of users conducting real research using your repository and its resources. Studies are not limited to one type of test or the collection of only one type of data.

Ongoing assessment

In addition to the data collected in formal studies, you will be collecting data from your users on an ongoing basis through use statistics, user feedback, and errors encountered by users. At the outset, do your best to provision your repository to be able to respond to this data. This includes setting up reliable feedback mechanisms users can employ and determining how you will respond to feedback. Having both a developer and designer assigned to support your repository will help you determine how and when to implement changes related to the feedback you receive. Tracking and responding to feedback data will fuel the ongoing improvement of your repository.

Conduct user studies and usability tests throughout the life of your repository using appropriate methodology.

Track and analyze passively collected data from your users.

Implement a plan for the processing of feedback into a sustainable design and development cycle.

Page Information

Date released: 23 Oct 2017
Last updated: 16 Oct 2017
Next review date: October 2018
Authors: Erin Faulder and Jennifer Colt-Demaree
Contact: Wendy Kozlowski, Repository Executive Group Chair (wak57@cornell.edu)
Related links:

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License