LD4 Knowledge Panel Recipe

Table of Contents

Introduction

“Knowledge panels” are an emerging feature in discovery applications (such as search engines, library catalogs and online shopping sites) that provide detailed information on a resource in an intuitive way to users. They give information service providers, such as libraries, a straightforward way of leveraging linked data to enrich their environments.

This paper seeks to roughly outline practical considerations and approaches to implementing knowledge panels in a technology-agnostic way. Additionally, common example uses and potential complications will be discussed. Its target audience is future implementers of knowledge panels using linked data.

This paper is a product of the LD4 Discovery Affinity Group as an effort to introduce linked data into library discovery environments. It was collaboratively produced by multiple contributors based on discussions at the Blacklight-LD Working Meeting at Stanford University in September 2019.

While these guidelines emerge from uses in the area of academic library discovery, many of these recommendations can be applied to any system where knowledge panels may help improve any website that references well-known concepts or entities.

Why are Knowledge Panels useful?

Key Points

- Knowledge panels are used by search engines such as Google to highlight and describe entities relevant to a search query.
- Multiple examples from libraries and museum discovery interfaces show the incorporation of some variation of a knowledge panel or supplemental information

retrieved from external sources. These examples cover both systems that are in production as well as those in prototype or experimental forms.

Definition of knowledge panel

Understanding what makes up a knowledge panel is most simply done by referencing their most prominent implementation in Google's search results page. For example, as shown in Figure 1, a Google search for "Octavia Butler" returns, in addition to links to external web pages, a section of the page on the right-hand side that includes photos, basic biographical data, and in this example, author-specific data such as authored works, quotations, similar authors and awards received.

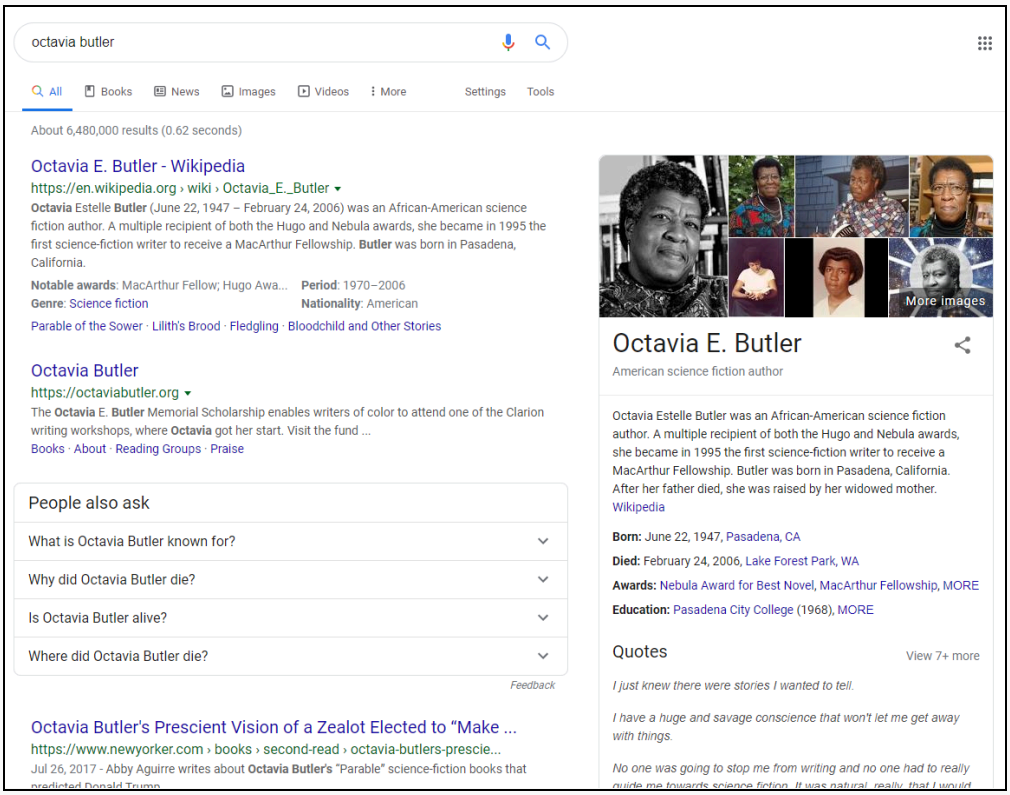

Figure 1: Screenshot of search for "Octavia Butler" (retrieved December 19, 2019)

The data displayed in Google's knowledge panel is likely not proprietary Google data, but drawn from public data such as Wikipedia. This points to the great value of knowledge panels - that you can integrate information from other sources into the content displayed by your site, adding value and enhancing your users' experience. But with the use of external data also come concerns about data quality, ethics and integrity, some of which will be discussed below.

Google knowledge panels tailor their presentation based on the type of entity being displayed as well as what information Google believes will be most relevant to the user. Geographic location information from the requesting user may also influence what results are returned. For example, as shown in Figure 2, searching for “myopia” will show a knowledge panel with relevant medical information for this concept. Searching for “Frank Sinatra” will display information such as birth and death date as well as genres which relate to Sinatra’s occupation as a musician.

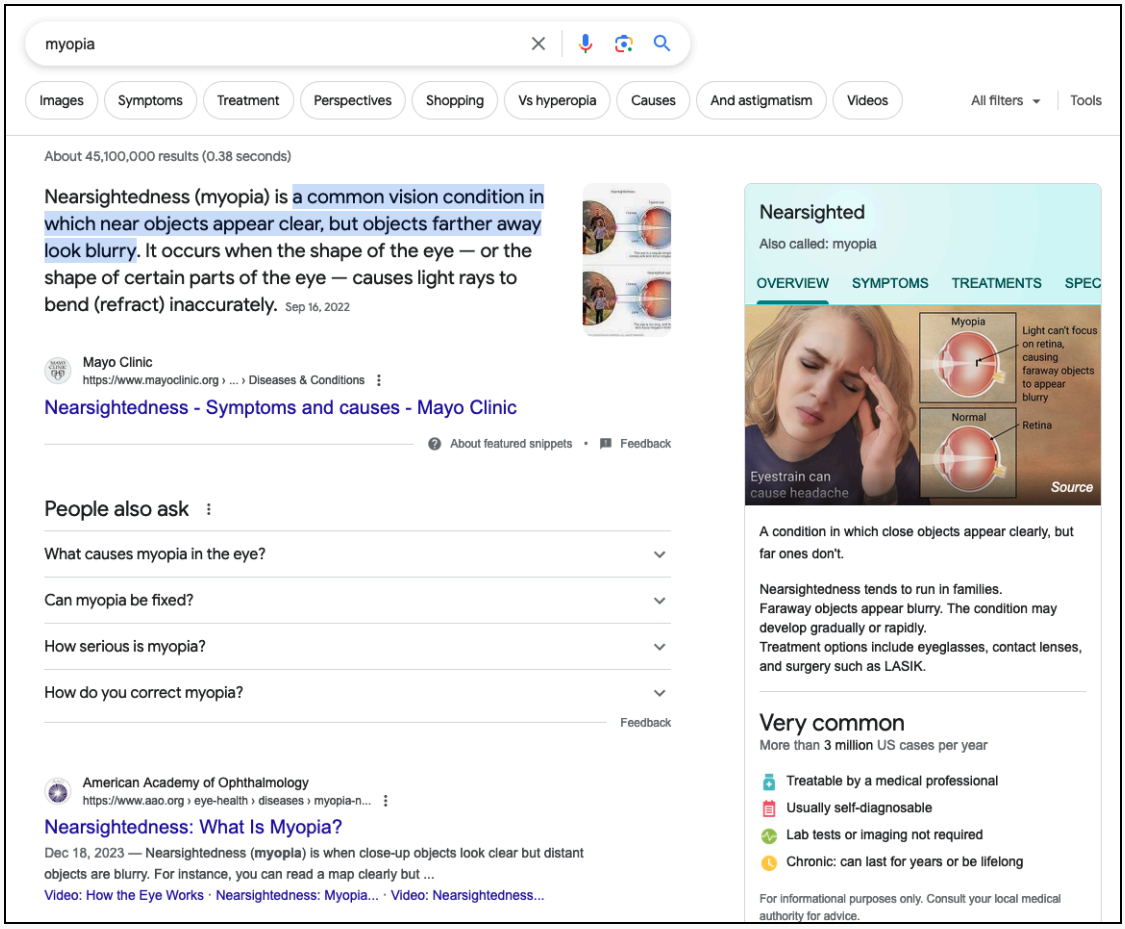

Figure 2: Searching for a medical concept returns different information in the knowledge panel than that for searching for a person (retrieved March 14, 2024)

As another example, Google determines whether a knowledge panel is included in search results based on a user's location. In Figure 3, a search of "carnegie library" performed from Ithaca, New York yielded a knowledge panel that includes information about the Carnegie Library at Syracuse University. The same search on the same day performed from Seattle, Washington (Figure 4) did not yield this knowledge panel.

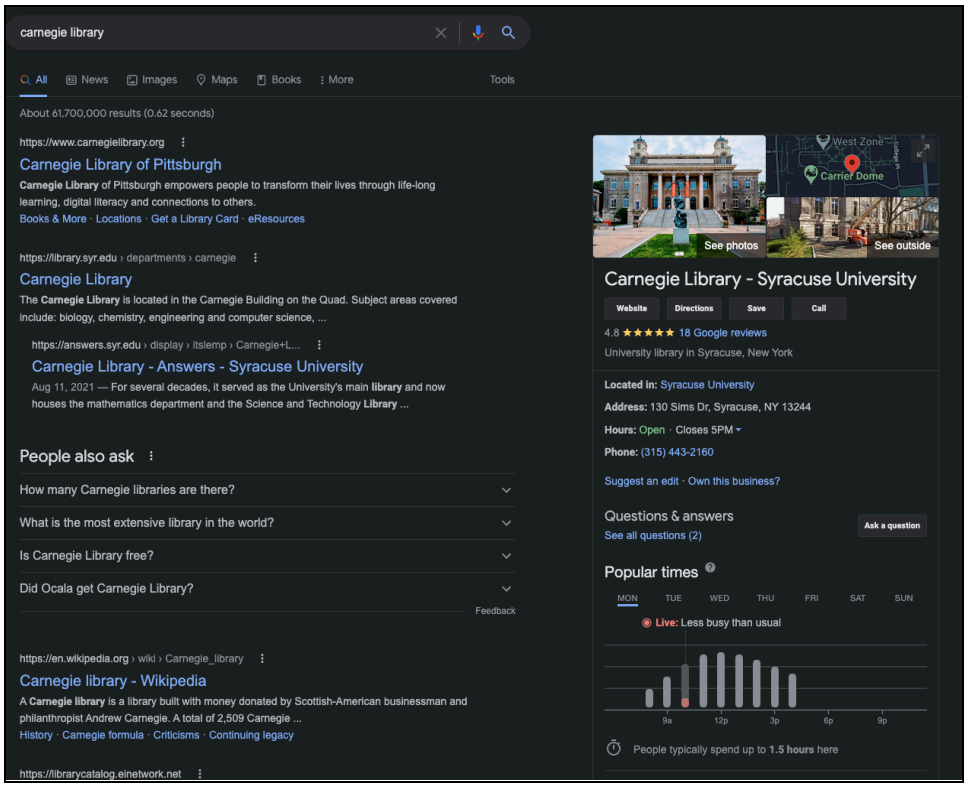

Figure 3: Screenshot of Google search for "carnegie library" from Ithaca, NY (retrieved December 20, 2021)

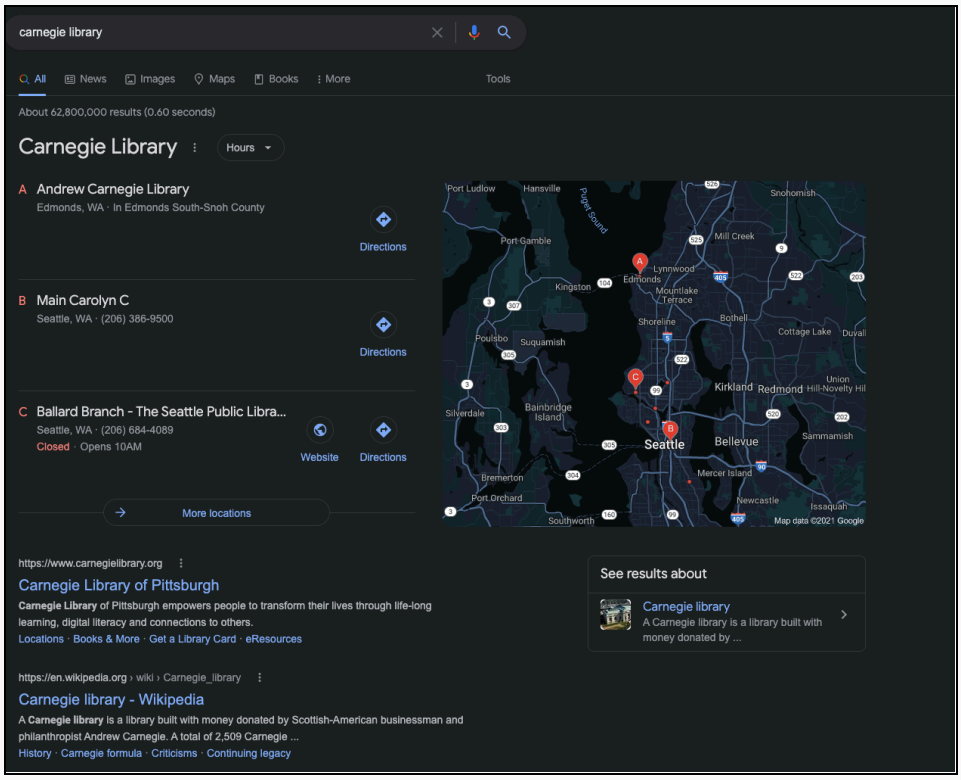

Figure 4: Screenshot of Google search for "carnegie library" from Seattle, WA (retrieved December 20, 2021)

Interactions made possible by the data returned

Linked data sources provide an amazing opportunity to incorporate information from non-library datasets into library discovery platforms. This extra information can provide context for the materials held by libraries such as providing biographical details that situate an author's historical relationship to the topics of the books being viewed in the library catalog. However, some of this information insofar as it is available as discrete data points can also provide more links in a catalog, affording new browsing opportunities for library patrons.

New interactions made possible by linked data derived knowledge panels may include links to information such as:

- Additional titles written by the subject of a knowledge panel,
- Additional people related to the author through professional or cultural association, and
- Titles for the sources cited by linked data assertions.

Biographical datasets, such as those common in sources derived from Wikipedia, often include properties like "notable works," which are defined by Wikidata as, "notable scientific, artistic or literary work, or other work of significance among subject's works."[1] For many authors, these will reference the same materials collected by a library, which make good candidates for linking across the catalog. In cases where the notable works describe either materials not owned by the library, these data still provide the user an expanded understanding of the subject.

For example, a knowledge panel for the contemporary painter Kehinde Wiley that displays in the UW-Madison library catalog lists his painting *Napoleon Leading the Army Over the Alps* as a notable work[2]. The keyword search resulting from the link in this knowledge panel finds a book on contemporary painters with a chapter devoted to this Wiley's painting[3]. In this case, the knowledge panel content matched against the bibliographic records transcription of the book's table of contents. In another example, the knowledge panel for the electronic musician Aphex Twin lists the album *Selected Ambient Works Volume II* as a notable work[4]. While the UW-Madison Libraries do not hold a copy of this particular album, they do have a book devoted entirely to the album. These examples serve to illustrate new opportunities for leveraging data available from linked data sources that can enhance discovery of library collections online.

Examples

Within the realm of libraries, museums, and archives, we have multiple examples of bringing in information to provide context around people related to a particular item, subjects, or geographic and spatial information. This information can be incorporated into the interface in multiple ways, from pop-ups that are visible when a user clicks on or hovers over a particular link, to information embedded in sections of the page.

The University of Wisconsin library catalog has incorporated contextual information from DBpedia, Getty, Wikidata and other sources into the page for a record in their library catalog (see Figure 5 below). As seen in Figure 6, Europeana displays information about subjects for an item and uses images from Wikimedia commons. Figure 7 shows how Michigan State University utilizes pop up knowledge panels to incorporate information from multiple sources around subjects and subject headings associated with a particular item. Laurentian University experimented with pulling in Wikidata links and representing this information in

[1] Wikidata. "notable work", accessed December 20, 2021: https://www.wikidata.org/wiki/Property:P800
[2] See “information from the web” on https://search.library.wisc.edu/catalog/9910116251802121
[3] See results of search on https://search.library.wisc.edu/search/catalog?q=Napoleon+Leading+the+Army+Over+the+Alps .
[4] See “information from the web” on https://search.library.wisc.edu/catalog/9910219289302121 .

knowledge panels. Cornell University created prototypes for knowledge panel pop ups that display information for authorized headings for contributors on an item's page. A later iteration of the Cornell University production library catalog employed knowledge panels for authors while also allowing users to click through to a detailed page displaying author information. The Library of the Pontificia Università della Santa Croce uses "authority boxes", which display authorized heading information in multiple knowledge panels with links to Wikipedia and WorldCat information. Additional examples with accompanying screenshots are available here.

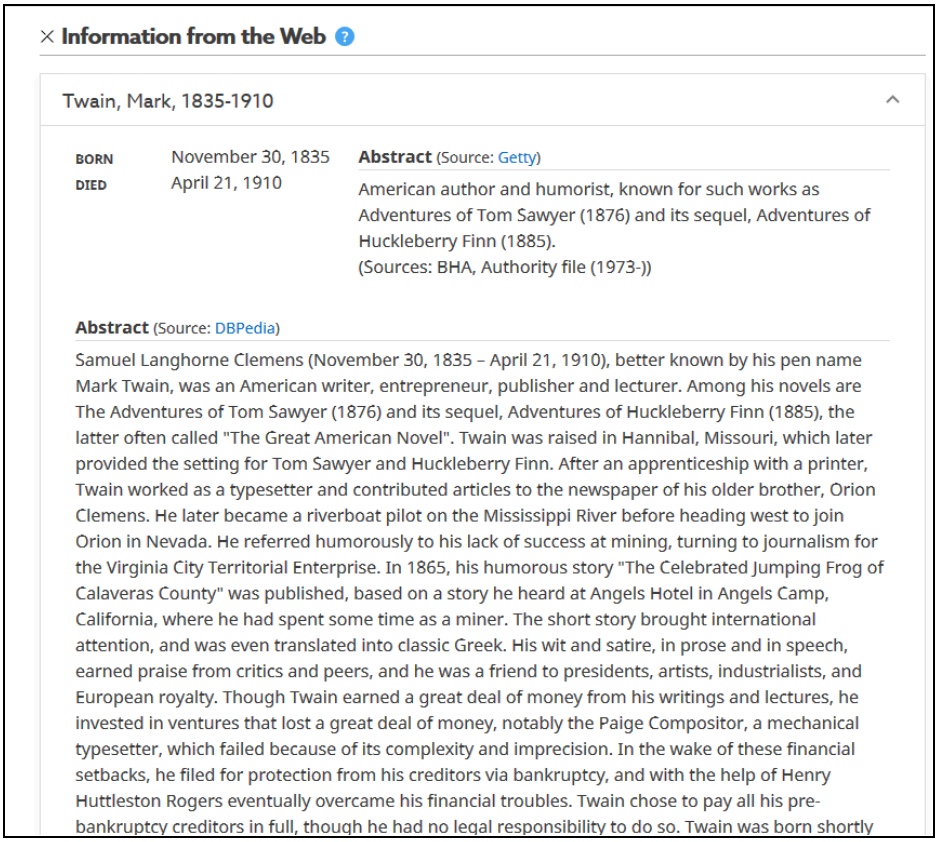

Figure 5: Screenshot of University of Wisconsin catalog displaying information about Mark Twain under a section entitled "Information from the Web" (retrieved November 25th, 2019)

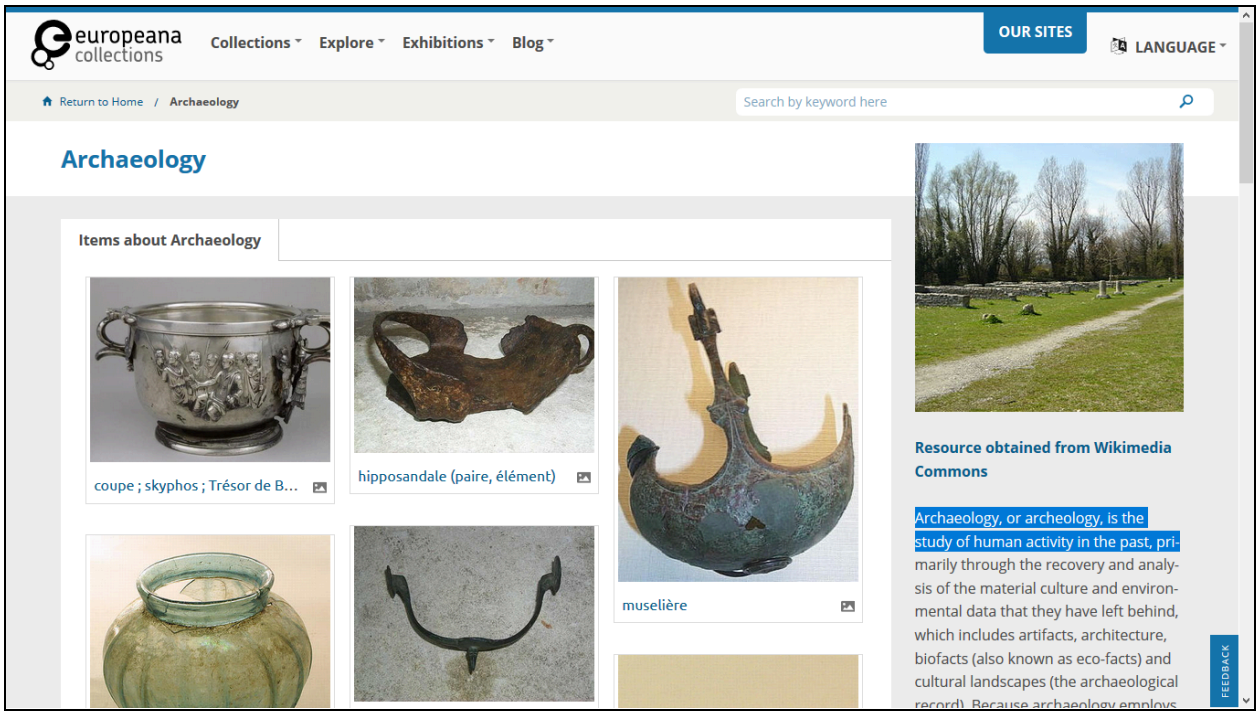

Figure 6: Screenshot from Europeana displaying images from Wikimedia related to the subject of Archaeology as well as a description for the subject (retrieved April 4, 2019)

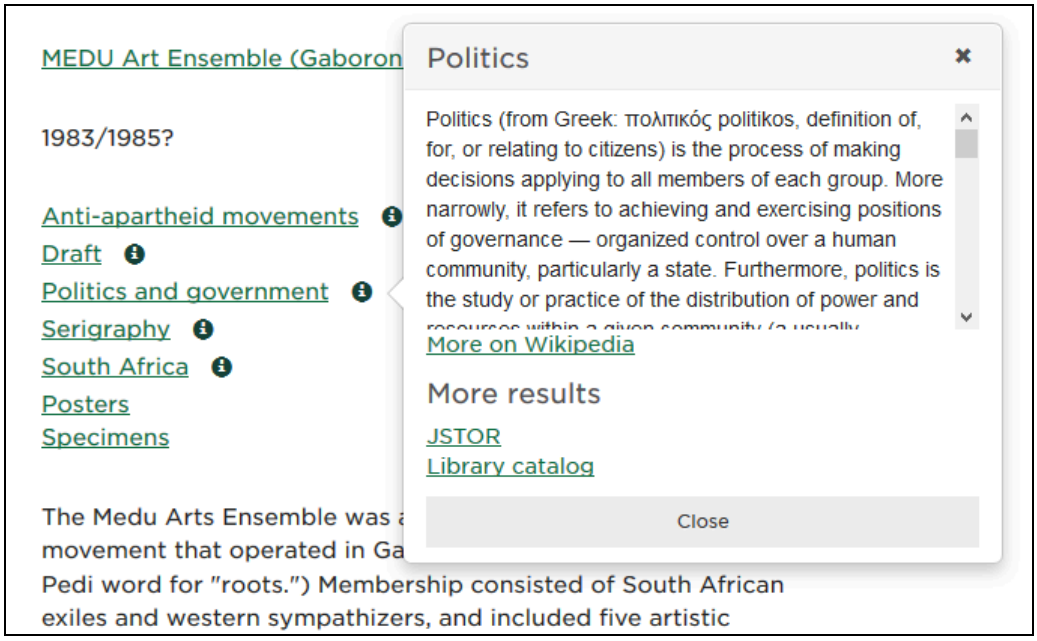

Figure 7: Screenshot of Michigan State University digital repository page for MEDU Art Ensemble, which includes pop-up knowledge panel for "Politics" subject heading that pull information from multiple sources such as Wikipedia (retrieved February 12, 2019)

In addition to systems like the ones above that are currently in production, different groups and institutions have undertaken experiments and investigations into the use of external data sources in discovery or cataloging systems. OCLC's Project Passage, although geared towards cataloging workflows, used discovery and contextual display interfaces that brought in information from DBPedia and WorldCat. As part of LD4P2[5] grant work, Stanford and Cornell have both created experimental systems that display knowledge panels incorporating information from external data sources. As seen in Figure 8, Stanford's Search Works LD prototype used information from Wikidata and Who's On First to show author and region knowledge panels. Figure 9 shows how Cornell's prototype used Wikidata as well as Library of Congress authorized headings in their knowledge panels for authors, subject headings, and narrative locations for items.

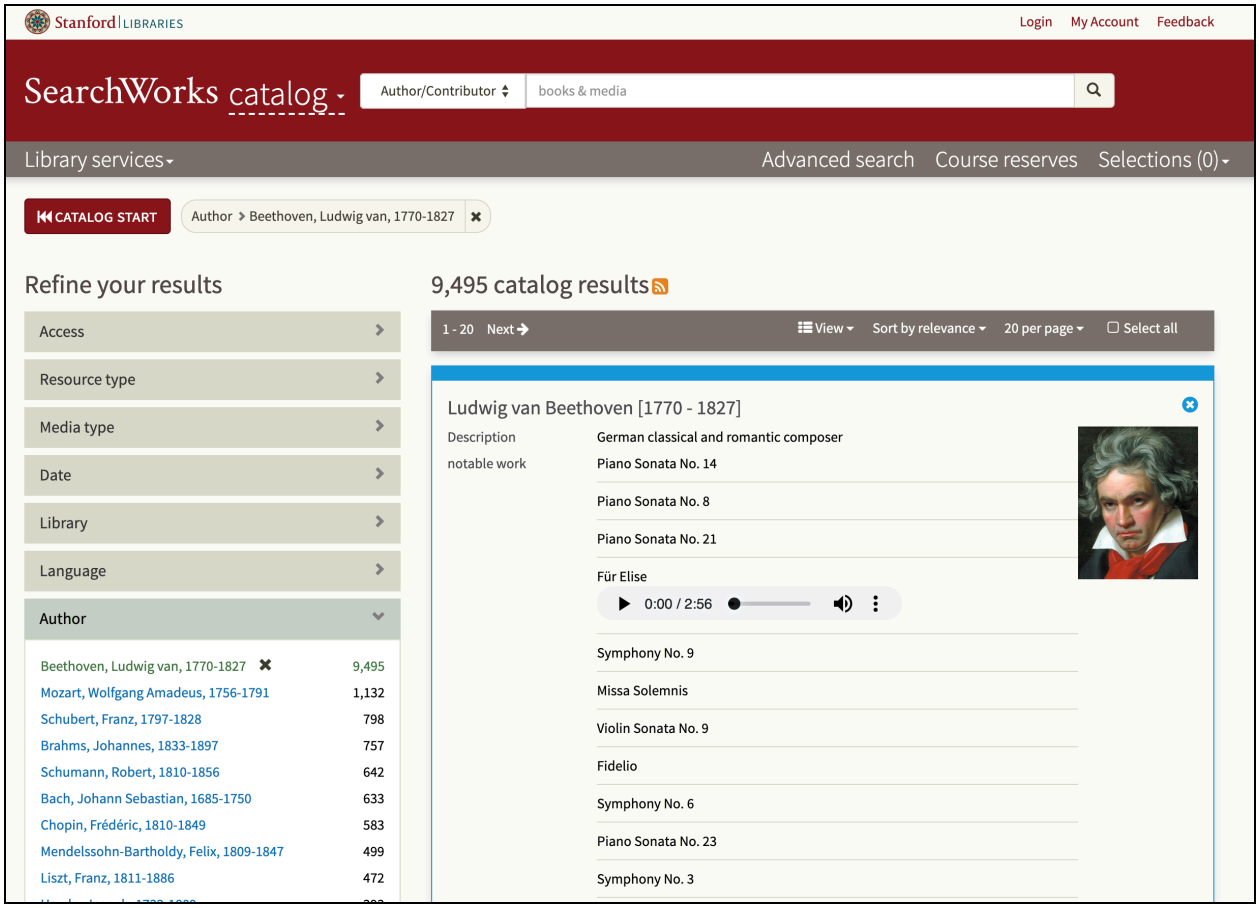

Figure 8: Screenshot of Stanford University SearchWorks-LD catalog with author knowledge panel in search results for author="Beethoven, Ludwig van, 1770-1827"*. Note this knowledge panel includes a sound file sourced from Wikidata (retrieved May 2019)*

[5] "Linked Data for Production: Pathway to Implementation (LD4P2)". https://wiki.lyrasis.org/display/LD4P2, accessed 2021-12-20

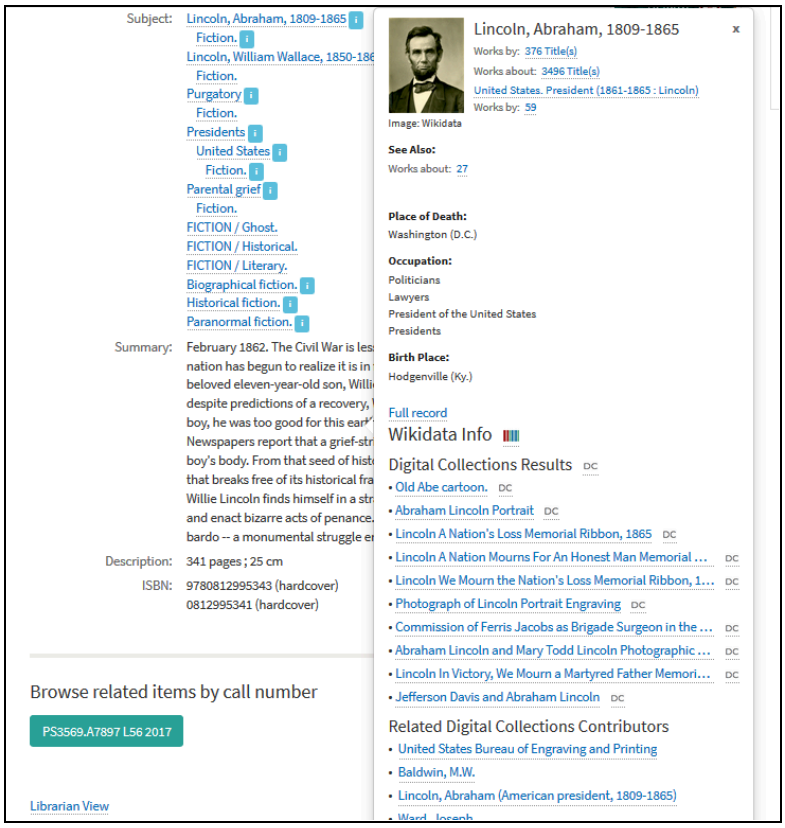

Figure 9: Screenshot of Cornell LD4P2 Knowledge Panel prototype integrating Wikidata information (retrieved 2019)

Data in library records

Key Points

- In order to query and retrieve information from linked data sources, the existing metadata needs to have Uniform Resource Identifiers (URIs) referring to those entities.
- There are multiple pathways for incorporating URIs into the discovery process, such as
    - Adding URIs directly to MARC bibliographic records either locally using lookup services or through vendor services.
    - Constructing URIs using authority control numbers

The primary requirement needed in order to surface a knowledge panel is the ability to construct a Uniform Resource Identifier (URI) so that one can retrieve data from another service. There are multiple pathways to construct a URI.

Probably the most common approach is to insert URIs into MARC bibliographic records. The MARC standard has two control subfields that can contain URIs. ǂ0, Authority record control number or standard number, and ǂ1, Real World Object URI. The referenced description says

> Subfield ǂ1 contains a Uniform Resource Identifier (URI) that identifies an entity, sometimes referred to as a Thing or Real World Object (RWO), whether actual or conceptual […] A URI that identifies a name or label for an entity is contained in ǂ0.[6]

There are a variety of ways one can insert URIs into MARC records. Existing catalog workflows can be enhanced to add the activity of looking up an authority in an identifier service and manually inserting the associated URI(s) into the appropriate subfield. Further, automated processes can look-up URIs via Application Programming Interfaces (APIs) and add those URIs into bibliographic records. Alternatively, one can utilize an authority vendor such as MARCIVE to insert URIs into vendor-supplied bibliographic records.

In addition to inserting URIs into MARC bibliographic records, one can construct a URI using authority control numbers from popular authority systems such as the Library of Congress Name Authority File (LCNAF) and Library of Congress Subject Headings (LCSH). For example, all one needs to do to construct a LCNAF URI via Library of Congress' Linked Data Service (id.loc.gov) is append the control number (ex. n2004123688) to a stable prefixed URL (http://id.loc.gov/authorities/names/ in this case). For more guidance, see the Program for Cooperative Cataloging Task Group on URIs in MARC's *Formulating and obtaining URIs: a guide to commonly used vocabularies and reference sources*.[7]

Many authority systems and community-based data sources (ex. Wikidata) build cross-references across URIs that represent the same entity. For example, the id.loc.gov record for Octavia Butler links to the Wikidata entity for Butler, which then links back to id.loc.gov as well as many other identifier services. In this case, one can resolve a single URI to retrieve all cross-referenced URIs to then pull in more URIs, thus utilizing a wider network of data.

Finally, there are a variety of reconciliation services for reconciling a list of uncontrolled text against an identifier service like Wikidata or id.loc.gov. OpenRefine has such a service built into the interface. Wikidata reconciliation is natively built in, and one can plug in other services like id.loc.gov.

---

[6] OCLC. (2024). *Control subfields*. Bibliographic Formats and Standards. https://www.oclc.org/bibformats/en/controlsubfields.html#subfield1, accessed 2024-03-14

[7] Program for Cooperative Cataloging Task Force on URIs in MARC (2020-01-15). "Formulating and Obtaining URIs: a Guide to Commonly Used Vocabularies and Reference Sources". https://www.loc.gov/aba/pcc/bibframe/TaskGroups/formulate_obtain_URI_guide.pdf, accessed 2021-12-20

The approaches described above can be used to insert URIs into source data like MARC, or they can be used to insert URIs into the data during indexing. One additional challenge to consider is that there may be more than one URI inserted into a source data record, such as MARC. These multiple URIs may reference a single entity (ex. an author), or they may reference different entities (ex. multiple authors for a given work). One should be conscious of this potential challenge and consult with metadata or cataloging professionals on workflows that can merge data from multiple URIs into a single panel or tease out the data for multiple resources into separate panels.

Data sources, APIs, and examples

Key Points

- This section provides specific examples of data properties available in particular linked data sources, queries that can be used to retrieve information for these properties, and the display of the retrieved information within the knowledge panel.

There are a variety of data sources that have been identified as potentially useful for building knowledge panels. Information about these sources has been aggregated in this spreadsheet[8]. There have also been past endeavors to identify and map specific properties from the data sources to entity types such as Person, Organization, Place, and Event. These examples are on this tab of the same spreadsheet[9].

[8] "Lord of the Rings Data Spreadsheet"
[9] "Info to include in knowledge panels" worksheet in "Lord of the Rings Data Spreadsheet" workbook

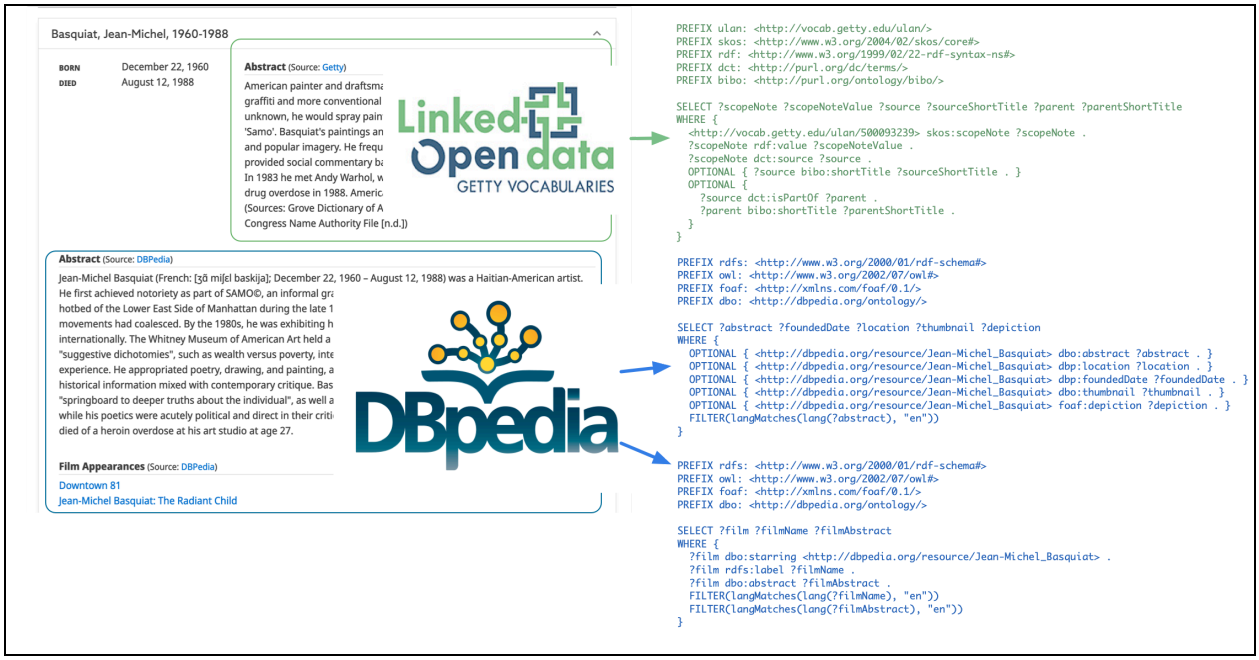

Figure 10: Illustrative example of knowledge panel with queries illustrating multiple sources

In the example presented in Figure 10, there are three data points highlighted coming from two linked data sources and retrieved using three SPARQL queries. The green section (upper-right) shows the biographical paragraph for Basquiat that was fetched from the Getty Vocabularies Linked Open Data. The second blue section (bottom-right) shows two data points from DBpedia: another biographical paragraph and a list of films that Basquiat appeared in. In the second example, two distinct queries were sent to DBpedia's SPARQL server.

As demonstrated in the Examples section above, Wikidata is a common dataset used for library knowledge panels. Below is an example profile of Wikidata properties that may be useful for adding context to Person names.

Property Name	Definition	Identifier
instance of	that class of which this subject is a particular example and member (subject typically an individual member with a proper name label); different from P279; using this property as a qualifier is deprecated—use P2868 or P3831 instead	http://www.wikidata.org/entity/P31
affiliation	organization that a person or organization is affiliated with	http://www.wikidata.org/entity/P1416

educated at	educational institution attended by subject	http://www.wikidata.org/entity/P69
employer	person or organization for which the subject works or worked	http://www.wikidata.org/entity/P108
notable work	notable scientific, artistic or literary work, or other work of significance among subject's works	http://www.wikidata.org/entity/P800
main subject	primary topic of a work (see also P180: depicts)	http://www.wikidata.org/entity/P921
partner in business or sport	professional collaborator	http://www.wikidata.org/entity/P1327
date of birth	date on which the subject was born	http://www.wikidata.org/entity/P569
date of death	date on which the subject died	http://www.wikidata.org/entity/P570
field of work	specialization of a person or organization; see P106 for the occupation	http://www.wikidata.org/entity/P101
occupation	occupation of a person; see also "field of work" (Property:P101), "position held" (Property:P39)	http://www.wikidata.org/entity/P106
work location	location where persons were active	http://www.wikidata.org/entity/P937
place of birth	most specific known (e.g. city instead of country, or hospital instead of city) birth location of a person, animal or fictional character	http://www.wikidata.org/entity/P19
place of death	most specific known (e.g. city instead of country, or hospital instead of city) death location of a person, animal or fictional character	http://www.wikidata.org/entity/P20
pronunciation audio	audio file with pronunciation	http://www.wikidata.org/entity/P443
image	image of relevant illustration of the subject; if available, use more specific properties (sample: coat of arms image, locator map, flag image, signature image, logo image, collage image); only images which exist on Wikimedia Commons are acceptable	http://www.wikidata.org/entity/P18

official website	URL of the official homepage of an item (current or former) [if the homepage changes, add an additional statement with preferred rank. Do not remove the former URL]	http://www.wikidata.org/entity/P856
Library of Congress authority ID	Library of Congress identifier for persons, organizations, events, places, titles, and subject headings [Format: 1-2 specific letters followed by 8-10 digits (see regex). For manifestations of works, use P1144]	http://www.wikidata.org/entity/P244

Table 1: Example profile of Wikidata properties that may be useful for adding context to Person names

There are a few ways to retrieve this information - all of these are listed here[10]. Wikidata URIs are dereferenceable, meaning that one can send an HTTP request for data formats such as JSON, RDF/XML, or TTL. There is also a SPARQL endpoint; examples of SPARQL queries to retrieve data from different kinds of entities are included in the following section.

SPARQL Examples for retrieving various properties, such as description, work location, occupation, employer, and affiliation for individuals

SPARQL Example: Occupation

The following query retrieves occupation information for Madeleine Ginsburg who is identified by the Wikidata identifier Q61139703.

```
SELECT ?occupation ?occupationLabel WHERE {
 wd:Q61139703 wdt:P106 ?occupation.
  SERVICE wikibase:label { bd:serviceParam wikibase:language
"[AUTO_LANGUAGE],en". }
}
```

.

The results of this query, in JSON format below, show us three different occupations: writer, curator, and editor. The Wikidata label service returns the label for the occupation in the occupationLabel variable.

```
[{"occupation":"http://www.wikidata.org/entity/Q36180",
```

[10] Wikidata. "Wikidata:Data access": https://www.wikidata.org/wiki/Wikidata:Data_access, accessed 2021-12-20

```
"occupationLabel":"writer"},
{"occupation":"http://www.wikidata.org/entity/Q674426","occupat
ionLabel":"curator"},
{"occupation":"http://www.wikidata.org/entity/Q1607826","occupa
tionLabel":"editor"}]
```

SPARQL Example: Description (bio), Work Location ID and Label for Tim Berners-Lee

```
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX pref: <http://www.wikidata.org/prop/reference/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?description ?workLocation ?workLocationLabel
WHERE
{
 <http://www.wikidata.org/entity/Q80> schema:description
?description .
 OPTIONAL {
      <http://www.wikidata.org/entity/Q80> wdt:P937
?workLocation .
 }
 SERVICE wikibase:label {
      bd:serviceParam wikibase:language "en" .
 }
 FILTER(langMatches(lang(?description), "en"))
}
```

The above query retrieves an English language description for the entity <http://www.wikidata.org/entity/Q80> representing Tim Berners-Lee along with work location information.

The above query will result in JSON including the following:

```
{"description":"British computer scientist, inventor of the
World Wide
Web","workLocation":"http://www.wikidata.org/entity/Q42944","wo
rkLocationLabel":"CERN"}
```

SPARQL Example for retrieving information for musicians and bands

The following query is taken from a description of the work done to populate musician and band knowledge panels in the Laurentian University catalog. This query returns information about a band, musician, or musical ensemble which has the label “A Tribe Called Red”. The Optional clauses below will return supplementary information such as birth place or instrument where it is available.

```
SELECT ?item ?itemLabel ?itemDescription ?image
 (GROUP_CONCAT(DISTINCT ?instrumentLabel;separator="; ") AS
?instruments)
 ?birthPlace ?birthPlaceLabel ?website ?musicbrainz
 ?songKick ?twitter ?facebook ?wplink
 WHERE {
    ?item rdfs:label|skos:altLabel|wdt:P1449 'A Tribe Called
Red'@en .
     # instance of = any subclass of band
    { ?item wdt:P31/wdt:P279* wd:Q215380 . }
     UNION
     # occupation = any subclass of musician
    { ?item wdt:P106/wdt:P279* wd:Q639669 . }
     UNION
     # instance of = any subclass of musical ensemble
    { ?item wdt:P31/wdt:P279* wd:Q2088357 . }
    OPTIONAL { ?item wdt:P3478 ?songKick } .
    OPTIONAL { ?item wdt:P19 ?birthPlace } .
    OPTIONAL { ?item wdt:P1303 ?instrument } .
    OPTIONAL { ?item wdt:P856 ?website } .
    OPTIONAL { ?item wdt:P434 ?musicbrainz } .
    OPTIONAL { ?item wdt:P2002 ?twitter } .
    OPTIONAL { ?item wdt:P2013 ?facebook } .
    OPTIONAL { ?item wdt:P18 ?image } .
    OPTIONAL {
     ?wplink schema:about ?item .
     ?wplink schema:inLanguage "en" .
     ?wplink schema:isPartOf <https://en.wikipedia.org/> .
    }
```

```
    SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en".
    ?instrument rdfs:label ?instrumentLabel.
    ?item rdfs:label ?itemLabel.
    ?item schema:description ?itemDescription.
    ?birthPlace rdfs:label ?birthPlaceLabel
   }
 }
GROUP BY ?item ?itemLabel ?itemDescription ?image ?birthPlace
?birthPlaceLabel
 ?website ?musicbrainz ?songKick ?twitter ?facebook ?wplink
LIMIT 10
```

SPARQL Example for retrieving wikidata ID based on LOC, VIAF, or ISNI URI (for Maya Angelou Q19526)

The following query can be used to take all the identifying URIs (in this case LOC, VIAF, and ISNI) and resolve to an entity Wikidata and return the Wikidata ID.

```
SELECT DISTINCT ?entity WHERE {
 { ?entity wdtn:P244 <https://id.loc.gov/authorities/n50024879> }
UNION
 { ?entity wdtn:P214 <https://viaf.org/viaf/7386077> }
UNION
 { ?entity wdtn:P213 <http://isni.org/isni/0000000121191991> }
}
```

SPARQL Example for retrieving literal values/identifiers/images/etc for an entity (for Maya Angelou Q19526)

The following query was cribbed from the SearchWorks linked data/Knowledge Panel Experiments [code]. The goal for this query is to get as much data as possible for a known wikidata entity and then allow the consuming code to filter the result to the attributes that they are interested in displaying.

```
PREFIX entity: <http://www.wikidata.org/entity/>
```

```
SELECT ?propUrl ?propLabel ?valLabel ?valUrl ?type
WHERE
{
 hint:Query hint:optimizer 'None' .
 {    BIND(entity:Q19526 AS ?valUrl) .
  BIND("N/A" AS ?propUrl ) .
  BIND("Name"@en AS ?propLabel ) .
  BIND("N/A"@en AS ?type ) .
  entity:Q19526 rdfs:label ?valLabel .
  FILTER (LANG(?val) = "en")
 }
 UNION
 {    entity:Q19526 ?propUrl ?valUrl .
  ?property ?ref ?propUrl .
  ?property rdf:type wikibase:Property .
  ?property rdfs:label ?propLabel.
  FILTER (lang(?propLabel) = "en")
  FILTER isliteral(?valUrl)
  BIND(?valUrl AS ?valLabel)
 }
 UNION
 {    entity:Q19526 ?propUrl ?valUrl .
  ?property ?ref ?propUrl .
  ?property rdf:type wikibase:Property .
  ?property rdfs:label ?propLabel.
  FILTER (lang(?propLabel) = "en")
  FILTER isIRI(?valUrl)
  ?valUrl rdfs:label ?valLabel
  FILTER (LANG(?valLabel) = "en")
  BIND('literal' AS ?type)
 }
 UNION
 {    entity:Q19526 ?propUrl ?valUrl .
  ?property ?ref ?propUrl .
  ?property rdf:type wikibase:Property .
  ?property rdfs:label ?propLabel.
  FILTER (lang(?propLabel) = "en")
  FILTER (?propUrl = wdt:P18 || ?propUrl = wdt:P154)
  BIND(?valUrl AS ?valLabel)
  BIND('image' AS ?type)
 }
 UNION
```

```
 { entity:Q19526 schema:description ?valLabel .
  FILTER (lang(?valLabel) = "en")
  BIND('Description' AS ?propLabel)
  BIND('fake-desc-uri' AS ?propUrl)
 }
 UNION
 { entity:Q19526 rdfs:label ?valLabel .
  FILTER (lang(?valLabel) = "en")
  BIND('label' as ?type)
  BIND('Label' AS ?propLabel)
  BIND('fake-label-uri' AS ?propUrl)
 }
}
ORDER BY ?propLabel
```

Implementation concerns: client-side and server-side solutions

Key Points

- Client-side solutions involve retrieval of information from an external source on page load or identified trigger event
    - Pros:
        - Distribute calls to users' IPs so linked data sources do not throttle requests
        - Newer frameworks incorporate API calls on the client-side
    - Cons:
        - Inability to cache activity for users
        - Could have a central module/section of the client-side handling API calls but that may still lead to making the code bigger
- Server-side solutions involve the application server making the calls to the external source. The client-side then communicates with the application server directly
    - Pros:
        - Caching possibilities
        - User anonymity (not passing user behavior to a third party).
        - Certain APIs/URLs may not be open for cross origin requests, which a server-side proxy can bypass
        - Certain services require keys, which can be configured at the server-side
        - Potentially easier for handling the differences between different source APIs than dealing with that on the client-side

        - Of note, in newer frameworks, the logic sits at the client-side. The routes/APIs are encoded into the client code. This may still be possible at client-side.
        - Possible normalization across different APIs
    - Cons/Considerations
        - If external data changes and the knowledge panel relies on the server, then external data updates need to be reprocessed/re-cached

Implementation Options: Client-side or server-side

When implementing linked data derived knowledge panels in a user interface, developers have two primary options. In both cases, code will be required to fetch linked data from one or more remote sources and build a displayable panel as a snippet of HTML. In the first option, this code exists on a web server. In the second, the code exists on the end user's client, typically a web browser. A client-side implementation provides the knowledge panel developer with the flexibility of remaining agnostic with respect to the backend solution used. A client-side knowledge panel solution would likely be written in the JavaScript programming language, which would also make it highly portable in multiple web application contexts. You might choose a server-side implementation if you need to have tighter privacy controls (see the Ethical Considerations section for discussion) or to add a data caching layer for large amounts of data aggregation.

Client-side implementation

In this technique, the hosting application provides the URIs of linked data entities via HTML markup and Javascript to the client (the user's web browser). The page loads, asynchronously queries the appropriate linked data endpoints, combines the information returned, and renders a knowledge panel with the desired content (using the Minimum Viable Display-ability rules). The panel may or may not be visible to the user at this point following a "Progressive Enhancement" pattern. This approach can be done in a framework agnostic way using technologies such as Web Components[11], which various application/discovery frameworks can then implement. Figure 11 provides an overview of this implementation approach.

One of the benefits to this approach is that the application server hosting the knowledge panel will not have to manage complexities such as throttling by the data source service as the requests will be distributed across the clients making the requests instead of the hosting server. One downside is the potential for performance implications given that it's not possible to efficiently cache data with this aspect. Users with poor connections or users requesting data from slow sources may see the supplementary data long after the rest of the page has loaded.

[11] https://www.webcomponents.org/introduction

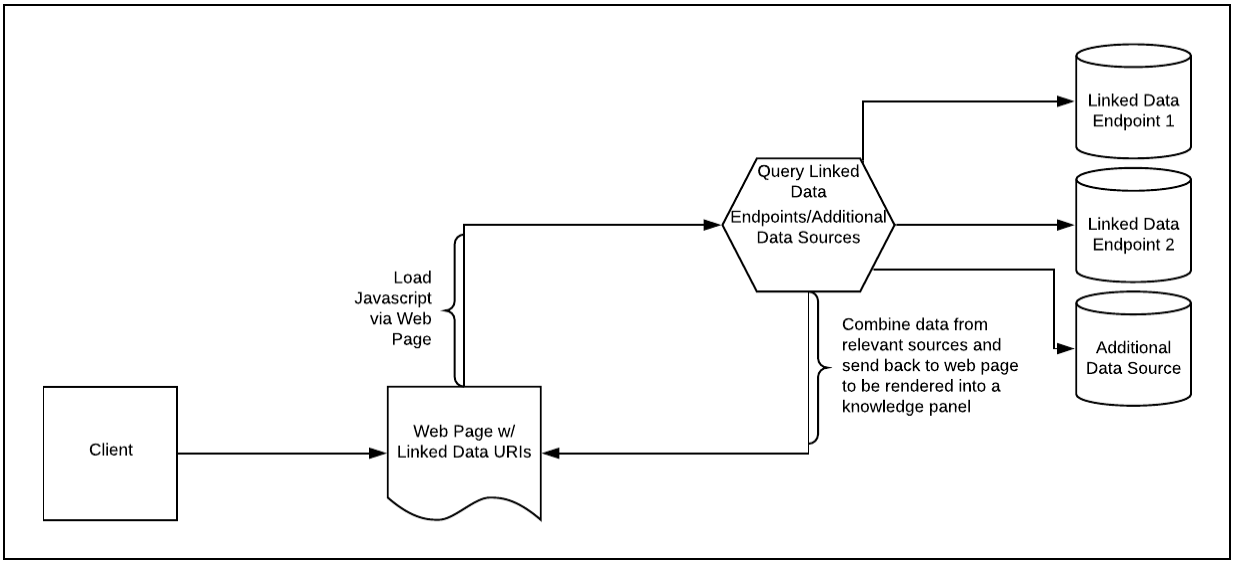

Figure 11 showing the basic interaction of the client-side implementation.

Server-side implementation

An alternative method for developing linked data derived knowledge panels involves a server-side solution. This technique involves an application server making calls to remote linked data sources. In this model, the application server assumes responsibility for communicating with linked data sources on behalf of a user's web browser. The primary technical benefit to this approach is the caching opportunities available when a web server acts as a proxy between a user's client web browser and the remote data sources. Caching data returned from the remote data sources enables the knowledge panel implementation to improve the user experience by providing faster load times. When data are cached , the application that provides a knowledge panel does not need to make repeated requests for the same information, thereby reducing the load on the data source.

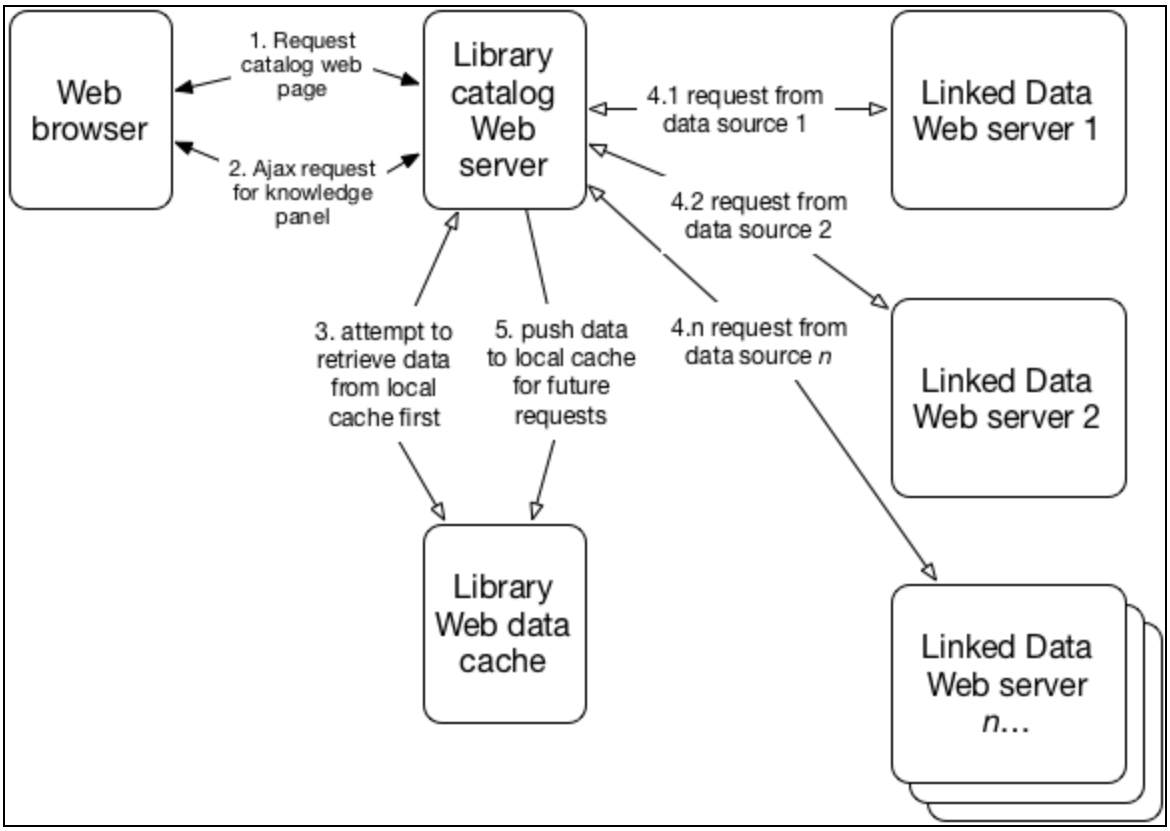

Figure 12: Illustration of data requests for a server-side knowledge panel implementation

Figure 12 illustrates a general design pattern for rendering knowledge panels in a library catalog using a server-side implementation. As noted above, a knowledge panel feature should be implemented in a manner that is non-disruptive to the core functions of a library catalog. This approach begins with an Ajax-based progressive enhancement to a web page that has already loaded most of its information. For example, the library catalog page describing a book should render bibliographic metadata first and then make an asynchronous call back to the web server to fetch a knowledge panel. In Figure 12 this first set of interactions is represented by the steps labeled 1 and 2. In step 2, the web page that displays a library catalog record will send back one or more identifiers, such as URIs, so the server knows which entities should be described in the resulting knowledge panel.

Next, the web server should attempt to retrieve the data that will be rendered in a knowledge panel from a local cache. If the data corresponding to the given entity is found, the catalog web application can render the knowledge panel as a small portion of HTML and respond to the browser's Ajax request. In this scenario, the process is complete as the knowledge panel will display in the user's web browser. In the case where the entity has not yet been cached or the cached copy of the data has expired, the catalog web application will fetch linked data from other remote sources online. In Figure 12, these are represented by step 4. Data from each source configured for use in the knowledge panel will be retrieved by direct HTTP requests against the original URIs or via queries. Once all data for a given entity are assembled, the final

step is to push a copy of the retrieved data for all sources into the web cache for subsequent requests (step 5 in Figure 12) just prior to rendering an HTML knowledge panel and fulfilling the web browser's Ajax request.

Under this model, the client-side of the web application is communicating only with the application server directly. Other technical benefits from a server-side approach include the ability to access content at remote endpoints that are not open for cross-origin requests and which would be blocked by web browser security features. Additionally, if any of the remote services required an authentication mechanism in the form of an API key, institutional credentials associated with a library will not be exposed to a client and published to a user's web browser.

The server-side implementation also has a few challenges. Namely, a library or organization providing functionality in this manner will need to support a web application infrastructure that can respond to the requests for knowledge card information, which would require local IT resources that may exceed the capacity of some institutions. Further, by requiring the organization to host a web server, this method will not work for organizations that use vendor products with limited access to installing add-on functionality. It is common practice for a vendor of discovery software to allow a library customer to inject custom JavaScript code into a web application to enhance the vendor's product. However, it is extremely unlikely that the vendor will host a server-side application developed by its customer. Finally, if the traffic becomes high, by funneling all requests for linked data through a single server with a single IP address, a remote data source may put throttling limits on the server and content may not be served up as quickly as if the IP addresses making the requests were distributed to individual user clients.

Pros/Cons Summary Table Across Implementation

	Purely client-side implementation	***Server-side implementation***
Data caching	Data caching not possible in a purely client-side implementation	Ability to cache data from external sources. If the entity had been previously requested, the cached version can be retrieved which may be faster and more reliable than a direct call to the external source. If employing caching, the application also needs to address keeping the cached information up to date with the external data source at a predetermined frequency.

Requests to data source	No requirement to address data throttling at external data source as requests distributed across clients	Requesting user's client-side information (e.g. IP address) is not shared with external source Central/single source for external data source requests Ability to request data from sources that may not allow cross-origin browser requests Knowledge panel implementation may face data throttling issues as external data source sees only one main source for all data requests made for that specific implementation.
Deployment	Code can be packaged in web components which can be integrated into any client-side solution without requiring additional server-side implementation.	Deploying additional server-side code may not be a preferred deployment step for many implementers especially in situations when using vendor solutions.

Table 2: Summary of pros and cons of client-side and server-side implementations

Data request performance considerations

Requests designed to occur concurrently occupy less total time than requests arranged serially. Knowledge panels may take advantage of this by making requests asynchronously except where one request is dependent on the outcome of a prior request.

For example, consider a knowledge panel creator who wants to display a list of related works compiled from two sources (e.g.: Wikidata and the Library of Congress) and who further wants to check whether each related work is present in the library catalog. Requests to both sources may be initiated at the same moment since requests for each source are not dependent on each other. Checking the results against the library catalog, however, can only begin when one or both data requests have finished.

A client-side implementation of this example could place asynchronous JavaScript AJAX requests to both Wikidata and the Library of Congress APIs. These requests will run with overlapping network wait times, collapsing both times into a single period that lasts only as long as the longer of the two. The return values of both requests can be combined, and duplicate results removed. This must occur after both requests are finished; thus, code would need to trigger a process at that time. JavaScript defines an object called a promise that makes this

possible. Each AJAX request may be written to return a promise, and further operations may be sequenced to execute when both promises are resolved.

Below is a code sample creating promises with the jQuery `$.ajax()` function, deferring code with `$.when.apply($, ...)` followed by `.done( ... )` until after the promises are resolved.

```
function ajaxRequestsForRelatedWorks(q) {
   var ajaxParametersList = [
      {
         url: 'https://lookup.ld4l.org/authorities/search/...',
         dataType: 'json'
      },
      {
         url: 'https://www.wikidata.org/w/api.php?action=...',
         dataType: 'jsonp'
      }
   ];
   // return an array of Ajax promises
   return ajaxParametersList.map(p => $.ajax(p));
}

function gatherRelatedWorks(q) {
   // get array of Ajax request promises
   var ajaxRequests = ajaxRequestsForRelatedWorks(q);
   // run each request in the array
   var whenRequests = $.when.apply($, ajaxRequests);
   // when done running, process responses
   whenRequests.done(function(ld4l, wikidata){
      // ... code to process request results here ...

      // pass the output of processing on to next function
      checkRelatedWorks(processedData);
   })
}
```

A server-side implementation of the example would use whatever techniques for asynchronous or concurrent requests were afforded by the server-side environment, allowing the client to request a unified list of related works.

In either client-side or server-side implementation, the developer may want to consider a timeout or cut-off that limits the length of time that may be occupied by a request. In the above example, the creator of the knowledge panel may decide that displaying results of one query for related works quickly is more valuable than displaying two queries if the two are more likely to increase time beyond a certain threshold. If so, the developer would implement a timeout discarding the longer-running of the queries, or perhaps both queries, beyond a certain number of seconds.

In some cases, an implementer must decide between retrieving information either as a series of simple requests, or consolidated into a single (or a smaller number) of more complex query requests. For an example, we will consider a knowledge panel on the subject of the computer scientist Tim Berners-Lee.

Berners-Lee is identified in Wikidata by the number Q80. All Wikidata statements that relate to this number can be found at the number's Wikidata URI. The cURL command line tool can dereference this URI like so:

```
curl -vL -H "Accept: text/plain" https://www.wikidata.org/entity/Q80
```

Among the many statements retrieved by this command, two of them relate to where Berners-Lee was educated:

```
<http://www.wikidata.org/entity/Q80> <http://www.wikidata.org/prop/direct/P69>
<http://www.wikidata.org/entity/Q73094> .
<http://www.wikidata.org/entity/Q80> <http://www.wikidata.org/prop/direct/P69>
<http://www.wikidata.org/entity/Q5369138> .
```

The reader will notice that this information is not human-readable. The institutions where Berners-Lee was educated are identified by the codes Q73094 and Q5369138, rather than by their names, Queen's College and Emanuel School. To display these names to the knowledge panel user, the URI for each school can in turn be dereferenced to get the display label.

This results in a pattern of making a request, seeing entities referred to in the result and making more requests to gather more information about each entity. This pattern has been called walking (or crawling) the graph.

Walking the graph may be an appropriate design choice in some circumstances, for example, when the need to dereference statements is conditional on a user interaction that occurs infrequently. Because there are thousands of Wikidata statements about Tim Berners-Lee in Wikidata, there is a practical limit to how many should be dereferenced in real time using this technique.

An alternative to walking the graph is to construct a SPARQL query. This SPARQL query retrieves the listing of Berners-Lee's education, together with the institutions' names, in a single request. It consolidates the work of several simple requests, but it is a more complex request. Queries to acquire information about Berners-Lee beyond his education may be more complex still.

Because SPARQL queries of this kind reduce the number of data requests, they may be the appropriate design choice in circumstances where walking the graph would result in too many requests; however, the developer should be aware that complex SPARQL queries can result in

Wikidata Query Service timeouts. This guide to query optimization[12] may be of use in addressing the problem.

Minimum Viable Display-ability

Key Points

- We propose the concept of Minimum Viable Display-ability to define the threshold for amount or categories of information needed before a knowledge panel is displayed.

In product and application development, the concept of a Minimum Viable Product[13] (MVP) defines the smallest feature set required to satisfy user needs for an application or product. The concept of MVP is useful because it defines the first benchmark that must be met before a product can be made available to its users. Once it is released into the world, a product can then be evaluated against its ability to solve the problems of real users rather than just the predetermined specifications of its designers. The insights that a design team can derive from real-world user evaluation can be very valuable to the early stages of a product development lifecycle by helping to identify areas of prioritization.

We would like to borrow the concept of MVP and apply it to knowledge panels, albeit with a slight twist. In our case we are proposing the concept of Minimum Viable Display-ability defined as the minimum amount of information required to display a sufficiently useful knowledge card. One of the challenges inherent in the development of knowledge panels is that it is often difficult to determine prior to query time whether or not the source datasets will contain information about the subject of a potential knowledge card. For example, in cases where fetching information about the author of a book does not yield any biographical information beyond what a typical bibliographic record has already recorded, such as birth and death dates that may be included in the authorized heading, it may not be useful to present a knowledge card to a library patron. The effect of presenting a knowledge panel that only displays the same information that is already displayed on the screen may make the feature confusing.

We are therefore proposing that implementers of knowledge panels test any data returned for a potential knowledge panel for display-ability. The University of Wisconsin-Madison has implemented a display-ability check before rendering a knowledge panel in their library catalog. A simplified version of their approach can be seen in Figure 13 below.

[12] Wikidata. "Wikidata:SPARQL query service / query optimization". https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/query_optimization, accessed 2021-12-20
[13] "Minimum viable product": https://en.wikipedia.org/wiki/Minimum_viable_product

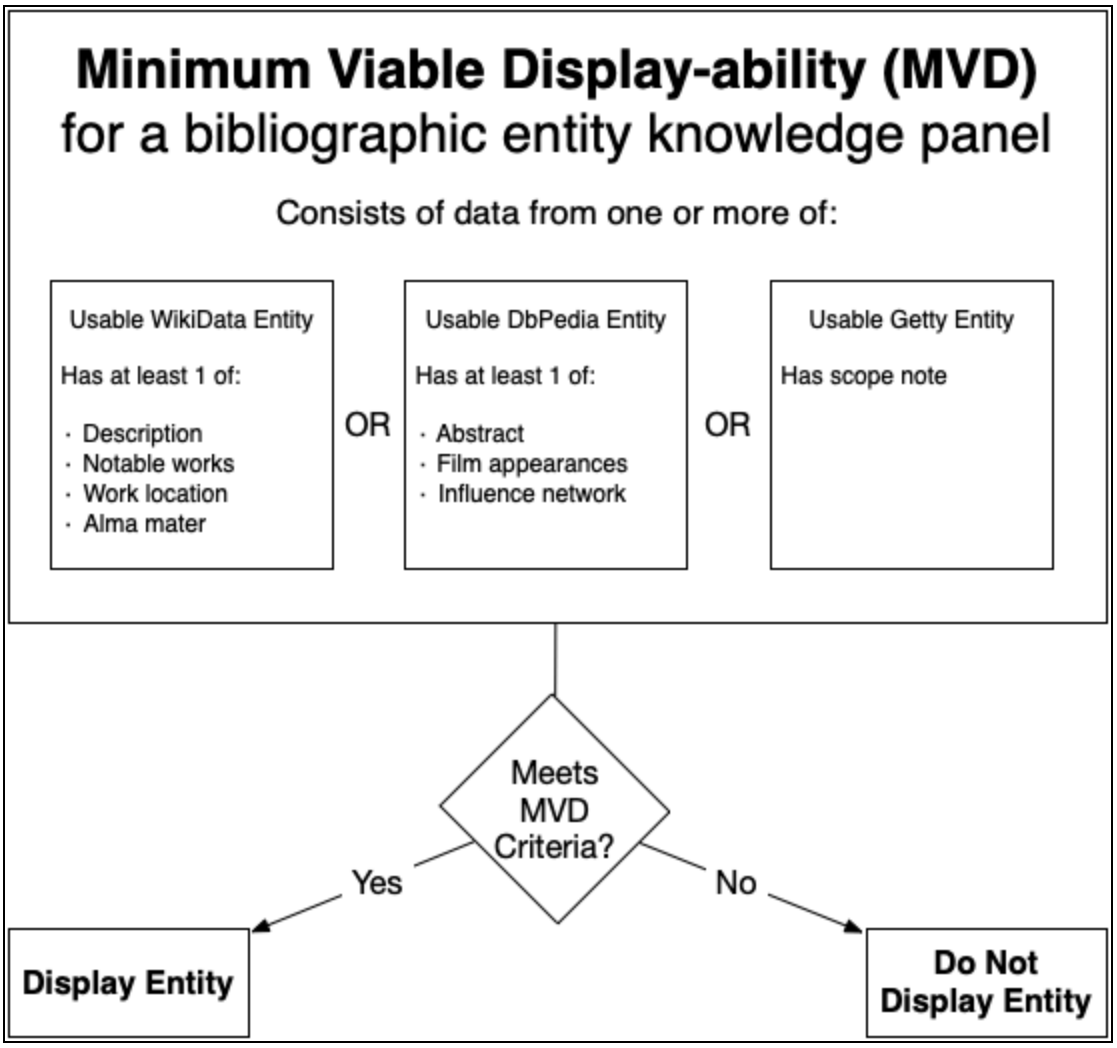

Figure 13: Example Minimum Viable Display-ability criteria.

The algorithm can be roughly translated as requiring that the resultant knowledge panel meets at least one of the following conditions:

1. A biographical abstract from one of three sources, or
2. A list of film appearances or related identities that influenced or were influenced by the subject, or
3. A list of notable works, a location in which the subject worked or a list of alma maters.

Each of the data points that define display-ability will contribute information to a knowledge panel that is not already available on a library catalog record. In this way, it can be seen that minimum viable display-ability may differ based on the particular use cases and contexts in which a knowledge panel resides.

Usability and User Experience Considerations

Key Points

- Knowledge panels may be integrated or displayed in response to different actions and in different sections of the interface. In this section, we review the different design choices possible for incorporating knowledge panels and how these choices may affect user experience and support.
- This section also includes a summary of user research and evaluation.

User research

Within the LD4P2 project, the project team conducted user interviews and evaluations of both mockups and functional prototypes to better understand whether knowledge panels may be useful, alongside what information is best to include in these panels. We have found that undergraduate students find knowledge panels useful mechanisms for providing relevant context for items in the catalog. In studies conducted with mockups[14] and the SearchWorks LD prototype[15] at Stanford, undergraduates favorably reviewed knowledge panels for authors incorporating information from Wikidata or Wikipedia. Author knowledge panels were triggered both when a user selected a particular author using a facet or when they hovered over an information button for contributors listed on the item view page. Additionally, geographic knowledge panels triggered when a particular region was selected in a facet were also considered potentially useful.

User evaluations[16] of knowledge panel mockups conducted at Cornell showed that participants were comfortable following links within knowledge panels to view related search results in the catalog. Participants had different perceptions regarding which particular properties were useful or how particular information was determined. For example, listing awards received and enabling discovery of items by authors who also received that award was met with mixed reactions, with some participants stating that knowledge of who won awards would be useful while others did not think this information would be beneficial. One participant wondered how the “influenced by” relationship was determined.

---

[14] Usong and Krogman (2019). "Undergraduate Student Discovery user research". https://docs.google.com/document/d/1LymSnnMm650jDmcrxx0BXaL7GRFQwVkVhEH-GP2FvE8/edit?usp=sharing, accessed 2021-12-20

[15] Usong and Doljack (2019). "May 2019 Knowledge Panel user research". https://docs.google.com/document/d/1CmN1DgDCNxGJBT3SA7Q63gN1zAJcgX22xXet6N0h-GI/edit?usp=sharing, accessed 2021-12-20

[16] Khan (2019-06-29). "KPAOW Mockups Feedback". https://docs.google.com/document/d/1BFWtki1LfmHsTtz6C25M9ZNv1GmJTN_VVg-fa34E6Ys/edit?usp=sharing, accessed 2021-12-20

Multiple factors can affect user experience with respect to knowledge panels; these include how and where knowledge panels or external information and context are added as well as the comprehensiveness and availability of the data itself. Ideally, the inclusion of additional context will not disrupt search and browse tasks currently supported by the interface. The model employed by Google where a single search result can be further highlighted in a knowledge panel may or may not work for every context or scenario. If we follow the Google model, we would need to ask which search result we should choose to highlight and why. If the data included in the knowledge panel is inconsistent or sparse, we may need to consider how much information or what types of information are present before we provide this option through the interface. As always, we should also be clear about the target audience for any design feature. As noted above, undergraduate students seemed much more amenable to the inclusion of these panels. Our research shows that more specialized researchers indicated they needed additional references and information than that which is usually displayed in knowledge panels.

Knowledge panel display

Data returned from sources may vary unpredictably in length with lists or arrays potentially including hundreds of related entities. To deal with this, the UI designer should consider affordances to expand and collapse displayed lists or to cut off list display beyond a certain number of items. Further, individual items in the list may be long; for example, a list of work titles may contain some titles with very high character counts. The UI designer may consider truncating titles with an ellipsis after a particular number of characters or pixels. On the other hand, data returned may be very short, or even empty. UI designers may consider suppressing the display of sections of a knowledge panel if the data returned is empty or too short to seem useful *(see section on Minimum Viable Display-ability)*.

Data sources that do not return unusually short or long responses during development cannot be relied upon not to produce them later. As a UI test, the developer may mock up unusual API responses, and may hunt for unusual responses by trying various queries. In doing so, the considerations for awkward data responses can be accommodated in the UI design.

Further, the developer cannot count on responses restricted to a familiar character set. The knowledge embodied by knowledge panels is worldwide; any language may occur in the results. The developer should be aware that data source APIs may restrict character sets of returned data based on the Content-Type, Accept, and Accept-Charset headers transmitted when the HTTP request is made, and that databases, browsers, and accessibility tools may differ in how they cope with various character ranges outside of the ASCII encoding system.

Even within English-language data results, punctuation marks present character encoding considerations. For example, quotation marks and apostrophes are sometimes encoded in so-called “curly” forms that can lead to character encoding mis-matches. However, the non-curly alternatives are significant characters in HTML, and must sometimes be escaped.

The guiding principle in crossing character sets is testing. The developer may seek out and mock up API responses of differing character sets as a UI test, just as they did to test responses of unpredictable length.

Integrity considerations

Key Points

- When using a data source, one needs to consider issues of how the data are maintained, who has control over the data, when and how data can be modified, and how often the data are changed; all of these factors have the potential to affect the accuracy and integrity of the data.

There are a few topics to consider before utilizing a dataset or data provider. First is currency - that is, how often is the dataset updated and how does that impact the accuracy/integrity of the data? For example, DBPedia, which is a major knowledge graph that is derived from Wikipedia documents, does not have a robust update schedule. In September 2017, the DBPedia page for the President of the United States was still listing Barack Obama as the president even though the DBpedia for Barack Obama correctly displayed the end date for his presidency. Currency of data is very important for datasets that are meant to encompass an ever-changing view of the world.

Ethical Considerations

In addition to the technical and functional considerations of introducing linked data in knowledge panels to your catalog, there are also ethical considerations, which share many problems with ethical issues in name authority records

These ethical concerns include:

- the use of women's maternal or married names (Kazmer 2019[17]; Martin 2019[18]; Whittaker 2019[19]; Olson and Schlegl 2013[20]),
- the use of colonial and anthropological nomenclature (Hughes 2019[21]; Elzi and Crowe 2019[22]; Biswas 2018[23]; Holloway 2018[24]; Vaughan 2018[25]),
- misspelled or miscopied non-Western names (Shiraishi 2019[26]; Arastoopoor and Ahmadinasab 2019[27]; Cohen 2019[28]; Whittaker 2019 (n 19)),
- racist terminology (Adler 2017[29]; Roberts and Noble 2016[30]; Antracoli and Rawdon 2019 [31]; Rigby and Gallant 2019[32]; Hughes 2019 (n 21))

---

[17] Kazmer, M. M. (2019). "Identify Theft: How Authority Control Undermines Women's Agency". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[18] Martin, J. M. (2019). "When Public Identity Meets Personal Privacy: Ethical Considerations for the Use of Dates of Birth in Name Authority Records for Living Persons". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[19] Whittaker, T. A. (2019). "Demographic Characteristics in Personal Name Authority Records and the Ethics of a Person-Centered Approach to Name Authority Control". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[20] Olson, H. A., & Schlegl, R. (2013). Bias in Subject Access Standards: A Content Analysis of the Critical Literature. *Proceedings of the Annual Conference of CAIS Actes Du congrès Annuel De l'ACSI*. https://doi.org/10.29173/cais460
[21] Hughes, H. K. (2019). "Cataloging Kurdistan: Imagining Liberated Geographies." In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[22] Elzi, E., & Crowe, K. M. (2019). "This is the Oppressor's Language Yet I Need It to Talk to You: Native American Name Authorities at the University of Denver." In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[23] Biswas, P. (2018). "Rooted in the Past: Use of 'East Indians' in Library of Congress Subject Headings." *Cataloging & Classification Quarterly, 56*(1): 1–18. https://doi.org/10/gg9cbm
[24] Holloway, S. W. (2018). "LCSH in the Southern Levant*." *Cataloging & Classification Quarterly, 56*(7): 571–91. https://doi.org/10/gg9cbn
[25] Vaughan, C. (2018). "The Language of Cataloguing: Deconstructing and Decolonizing Systems of Organization in Libraries." *Dalhousie Journal of Interdisciplinary Management, 14* (Spring): 1–15
[26] Shiraishi, N. (2019). "Accuracy of Identity Information and Name Authority Records". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[27] Arastoopoor, S., & Ahmadinasab, F. (2019). "From Personal to Corporate and from Names to Titles: the Challenges of Iranian Scholars with Scientific Publications". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[28] Cohen, A. (2019). "Free to Be… Only He or She: Overcoming Obstacles to Accurately Recording Gender Identity in a Highly-Gendered Language". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[29] Adler, M. (2017). "Classification Along the Color Line: Excavating Racism in the Stacks." *Journal of Critical Library and Information Studies, 1*(1). https://doi.org/10/gdb8q3
[30] Noble, S. U., & Roberts, S. T. (2016). "Through Google-Colored Glass(es): Design, Emotion, Class, and Wearables as Commodity and Control". In S. Tettegah & S. Noble (Eds.), Emotions, Technology & Design. pp. 187-210. San Diego: Elsevier Academic Press.
[31] Antracoli, A. A., & Rawdon, K. (2019). "What's in a Name? Archives for Black Lives in Philadelphia and the Impact of Names and Name Authorities in Archival Description". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.
[32] Rigby, C., & Gallant, R. (2019). "Creating Multilingual and Multiscript Name Authority Records: A Case Study in Meeting the Needs of Inuit Language Speakers in Nunavut". In J. Sandberg (Ed.), *Ethical Questions in Name Authority Control*. Sacramento, CA: Library Juice Press.

- issues in the names and pronouns of transgender, transsexual, gender nonconforming, or authors of nonbinary identity (Tanenbaum 2020[33], Wood 2019[34]; Beemyn 2019[35]; Sinclair-Palm 2017[36]; Marine and Nicolazzo 2014[37]; Rawson 2018[38])
- The use of a groups preferred terminology (A4BLIP 2019[39], Sigelman, Tuch, and Martin 2005[40])

Understanding who maintains the dataset or who the data provider is will shed light on the integrity of the dataset and may raise potential concerns to address. For example the LCNAF is maintained by the Library of Congress and data are contributed by members of the Name Authority Cooperative Program (NACO), which is a program of the Program for Cooperative Cataloging (PCC). In short, there is a rigorous process in place to contribute data to LCNAF that ensures authoritativeness of the data; it is important to remember that authoritativeness does not guarantee accuracy. It's a similar setup for Library of Congress Subject Headings (LCSH); however, there is a laborious process for proposing changes to records already in these national authority files. There is a wide acknowledgement from the library community that there are terms in these authority files that are outdated or offensive, especially terms that represent marginalized communities. Proposals to change these problematic terms have to go all the way through the policy committee of the Library of Congress.

To contrast this, Wikidata is a community-based knowledge graph; this means that - anyone (with an internet connection) can contribute to Wikidata. Because of this, Wikidata is easier to update and draws upon the power of the community to maintain comprehensiveness and integrity. However, where Wikidata gains strength from robustness, it falls short on authority. There have been instances of bias and bad actors that have disrupted its integrity. There are communities like WikiProject Counter-Vandalism and tools for addressing vandalism in

---

[33] Tanenbaum, T. J., Speer, R., Rettig, I, Goetz, T. G., Toups Dugas,P. O., Spiel, K., & Watson, B. M. (2020). Towards a trans inclusive publishing landscape. Medium. https://medium.com/the-name-change-policy-working-group/towards-a-trans-inclusive-publishing-landscape-893339b9868d, accessed 2024-03-14

[34] Wood, K.. 2019. "Whose Archives? Legal and Ethical Considerations in Provenance, Ownership and Responsibility." *Law & History, 6*(2): 9–27.

[35] Beemyn, ed. 2019. Trans People in Higher Education. Albany, NY: SUNY Press.

[36] Sinclair-Palm 2017. "'It's Non-Existent': Haunting in Trans Youth Narratives about Naming." *Bank Street Occasional Paper Series, 37* (Queering Education: Pedagogy, Curriculum, Policy): 7.

[37] Marine, S. B., & Nicolazzo, Z.. (2014). "Names That Matter: Exploring the Tensions of Campus LGBTQ Centers and Trans* Inclusion." *Journal of Diversity in Higher Education, 7*(4): 265–81. https://doi.org/10/gf3fpq

[38] Rawson, K. J. (2018). "The Rhetorical Power of Archival Description: Classifying Images of Gender Transgression." *Rhetoric Society Quarterly, 48*(4): 327–51. https://doi.org/10/gfkhvc

[39]Archives for Black Lives in Philadelphia (A4BLiP) Anti-Racist Description Resources: https://www2.archivists.org/standards/archives-for-black-lives-in-philadelphia-a4blip-anti-racist-description-resources

[40] Sigelman, L., Tuch, S. A., & Martin, J. K. (2005). What's in a Name? Preference for "Black" versus "African-American" among Americans of African Descent. *Public Opinion Quarterly, 69*(3), 429–438. https://doi.org/10.1093/poq/nfi026

Wikidata, but it's an imperfect system. Wikidata contains rich data that are valuable for adding context to library discovery systems, but one needs to be mindful of potential challenges that are presented from such a community-based dataset, and build policies/strategies for acknowledging and addressing these challenges. For example, the UW-Madison Libraries have been piloting adding context to their catalog names (authors, contributors, name-subjects etc) using Wikidata, DBpedia and Getty's Linked Open Data. They initially were utilizing images from DBpedia until they came across the image for Chelsea Manning, which, at least at that point in time, was an image of her before her transition. After consideration of the careful and thoughtful analysis by Matthew Reidsma[41] (Grand Valley State University), and Ruth Tillman (Penn State University Libraries), library staff were concerned about the potential but serious harm that could be caused by the UW-Madison Libraries catalog display of the outdated photograph. As a result, the knowledge panel image display feature was disabled. For more information on the University of Wisconsin's impressively-thorough effort of assessing knowledge panels--including from an ethical perspective--from a staff and patron perspective[42], see their Blacklight LD Summit presentation.

An additional ethical concern for building knowledge cards concerns whether a library's application shares its user browsing activities with third party data sources. For example, one motivation for implementing a knowledge panel feature using the server-side approach described above is that all requests for data are proxied by the library's application server. When requests for data are made by an application server, information such as a user's IP address are never directly shared with a third party data source. Instead the third party data service only sees repeated requests for information coming from a single source, such as the web server hosting a library catalog. This technique will therefore contribute to the mission of a library that wishes to minimize harm by respecting a patron's right to privacy as described in the American Library Association's Library Bill of Rights: "All people, regardless of origin, age, background, or views, possess a right to privacy and confidentiality in their library use. Libraries should advocate for, educate about, and protect people's privacy, safeguarding all library use data, including personally identifiable information."[43]

One recommended practice emerging from data science fields is to develop and apply a "data checklist" for your institutional use of linked data. Just as pre-flight checklists can reduce accidents, and medical checklists can reduce accidents in surgery, a data checklist can help you avoid mis-using data. The LD4 Ethics in Linked Data Affinity Group developed such a checklist, which is divided into planning, implementation, and maintenance sections. Each section covers questions and prompts on the following topics: general; accessibility; provenance, sources, and

[41] Reidsma, M. (2019-03-05). "Turning off Summon's Topic Explorer sidebar". Work Notes. https://matthew.reidsrow.com/worknotes/214 , accessed 2021-12-20
[42] UW-Madison Libraries. (2019). "Staff & Patron Assessment of Knowledge Panels". From Blacklight LD Workshop. https://docs.google.com/presentation/d/1bHf-B3sxePzRK_T89dyk2jGjpARTOJ9w/edit?usp=sharing&ouid=104226061230195732602&rtpof=true&sd=true, accessed 2021-12-20
[43] http://www.ala.org/advocacy/intfreedom/librarybill

citations; oppression and harm; inclusion and diversity; identity management and privacy; and data sovereignty and intellectual property[44].

Making a Business Case to Implement Knowledge Panels at your Institution

Making a successful business case for the implementation of knowledge panels requires balancing the value to be added with the work required (for catalogers and web developers) and the risks introduced (realistically conceived). Below are some points to consider in whether or not implementing knowledge panels may be worthwhile for your individual use cases.

Costs/Risks & Mitigation/Opportunities

There are valid concerns about costs and risks with implementing knowledge panels that rely on external linked data. The table below is meant to acknowledge these risks and suggest mitigation strategies and opportunities that come with adoption.

Cost/Risk	Mitigation/Opportunity
External data is not authoritative	In some cases, the external data is more authoritative than the data produced by people with domain knowledge. We can incorporate display and search controls in our systems that consume this external data, so we can decide which data to display or use and which data to ignore.
External data is not accurate	Offer contextual “Correct this on Wikidata / Wikipedia” links, if appropriate, otherwise offer contextual “Report incorrect data” links. We can incorporate display and search controls in our systems that consume this external data, so we can decide which data to display or use and which data to ignore. We can benefit from the wisdom of the crowd and enrich our catalogs with expert data beyond our institutions.
External data is not comprehensive	Library data isn’t necessarily comprehensive in all

[44] Ethics in Linked Data Affinity Group (2023). Ethics in Linked Data Checklist. In Watson, B. M., Provo, A. A., & Burlingame, K. (Eds.). *Ethics in Linked Data*. https://doi.org/10.5281/zenodo.10258209

	cases. Incorporating external data has proven useful in helping patrons disambiguate between different resources, creators, etc. with added context when our internal records are sparse or otherwise unclear. Similar to the above point, we can decide which external data to utilize.
Patrons will not understand or will not like it	Institutions that have experimented with linked data or have incorporated that data into their production systems have focused on designs that support and improve user experience. Patrons may find non-library linked data more approachable than library classifications. (See UWisconsin, Cornell, Stanford Mockup testing and Prototype testing results).
It's not useful to patrons; (or it's only useful for superficial needs)	The usability and user research references above show that patrons find context and additional metadata useful in determining the relevance of library resources. External data can also allow patrons to make connections and browse between resources where keyword searching is not getting them to the most applicable resources, or enable serendipitous discovery.
Library staff won't understand or like it	We can provide better training and support to help library staff in understanding linked data and its possible uses. In addition, library staff can engage in the design and development of the knowledge panels, and feel comfortable with the outcome. It is also important to note that non-library linked data may be more approachable and easier to understand than some library standards. While there is a learning curve for library technologists, major players on the web are engaging with linked data and semantic technologies, making adoption a progressive strategy.
It's expensive and hard to implement	This white paper makes it easier; there is reusable code. Some commercial systems, like Google, already implement knowledge panels. Leveraging open data is an extension of "copy cataloging" and acknowledges we can't afford to capture all the necessary data ourselves.
It's new; we haven't done this before	At the current time, some libraries, such as University of Wisconsin at Madison and Cornell University, have

	already implemented knowledge panels, so library knowledge panels are not that new anymore.
Why not just link to data, and let users find it outside the catalog?	Less clicks to see data: Cornell, Stanford Mockup testing and Prototype testing Ok to allow people to see info outside the catalog, but using links to provide context for catalog resources should help catalog discovery, and also make the catalog stickier

Table 3: Costs and mitigations for implementing knowledge panels using external data sources