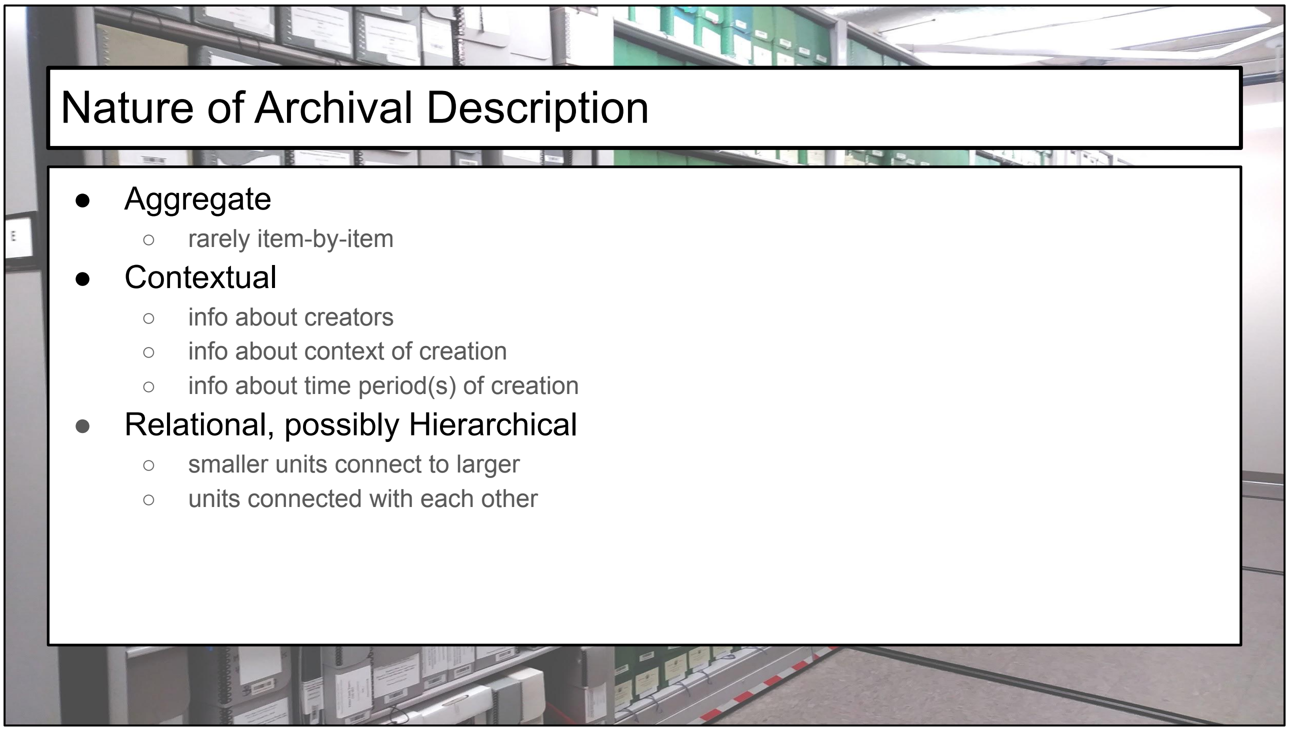
Hollinger's Box: The Retrieval Object at the Edge of the Ontology

Regine Heberlein & Ruth Tillman

Good afternoon everyone. We are Ruth Tillman and Regine Heberlein and our presentation title is "Hollinger's Box: The Retrieval Object at the Edge of the Ontology."



Nature of Archival Description

- Aggregate
  - rarely item-by-item
- Contextual
  - info about creators
  - info about context of creation
  - info about time period(s) of creation
- Relational, possibly Hierarchical
  - smaller units connect to larger
  - units connected with each other

With this presentation, we hope to offer a functional review of three linked data models for archival description.

We don't assume that everyone here is familiar with archival description. An archival collection or fonds may contain thousands or millions of individual items – letters, forms, memos, photographs, etc. These items don't declare themselves the way a published resource does: they generally don't have a title, frequently don't have a date or attribution and so on. What they *are* is inferred from the archival Context in which they have been maintained. An address list found within the donor files of, say, presidential campaign files can be inferred to be the contacts of donors to the presidential campaign even though the piece of paper may be neither titled nor dated.

Archival description provides researchers with an overview of what record groups they can expect to find in a collection. It generally includes information about the Context in which these records were created – who created them? Were they performing a specific function, such as university registrar, campaign manager, donor? Over what time period were they created?

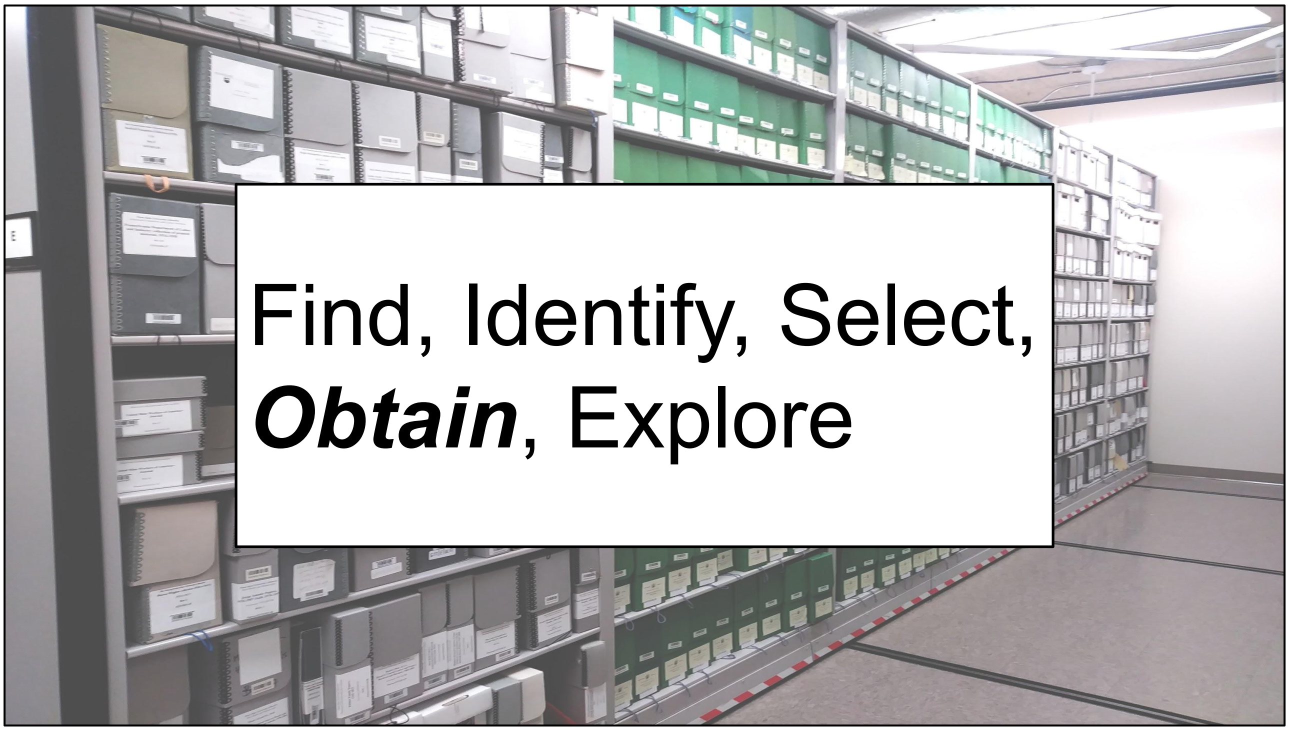
These aggregate descriptive units in turn relate to other units. The letters of the university registrar, for example, would nest under and inherit properties from an overall description of the registrar's records. When we move beyond the traditional hierarchical document format, however, we see connections to other aggregates, such as to all university correspondence. This is the promise of linked data for archival description.

Linked Data and Retrieval

Aggregate description has consequences for archival retrieval that set it apart from bibliographic retrieval. In the bibliographic domain, the retrieval object is typically the item: a volume. By describing the item and adding locator information to that description, we have enough information to go find the book on a shelf, on a floor of a building, at a library etc.

The *archival* retrieval object on the other hand is more typically a nested hierarchy of containers and subcontainers, with the subcontainers holding multiple items…

…so if for example if you wanted to consult Edith Wharton's 1925 note inviting the Fitzgeralds to tea, someone would retrieve Box 49 of the F. Scott Fitzgerald Papers from Princeton University's closed stacks and deliver it to you at the reading room in the main library building. You would find the note in Folder 41, along with other correspondence between Wharton and Fitzgerald.

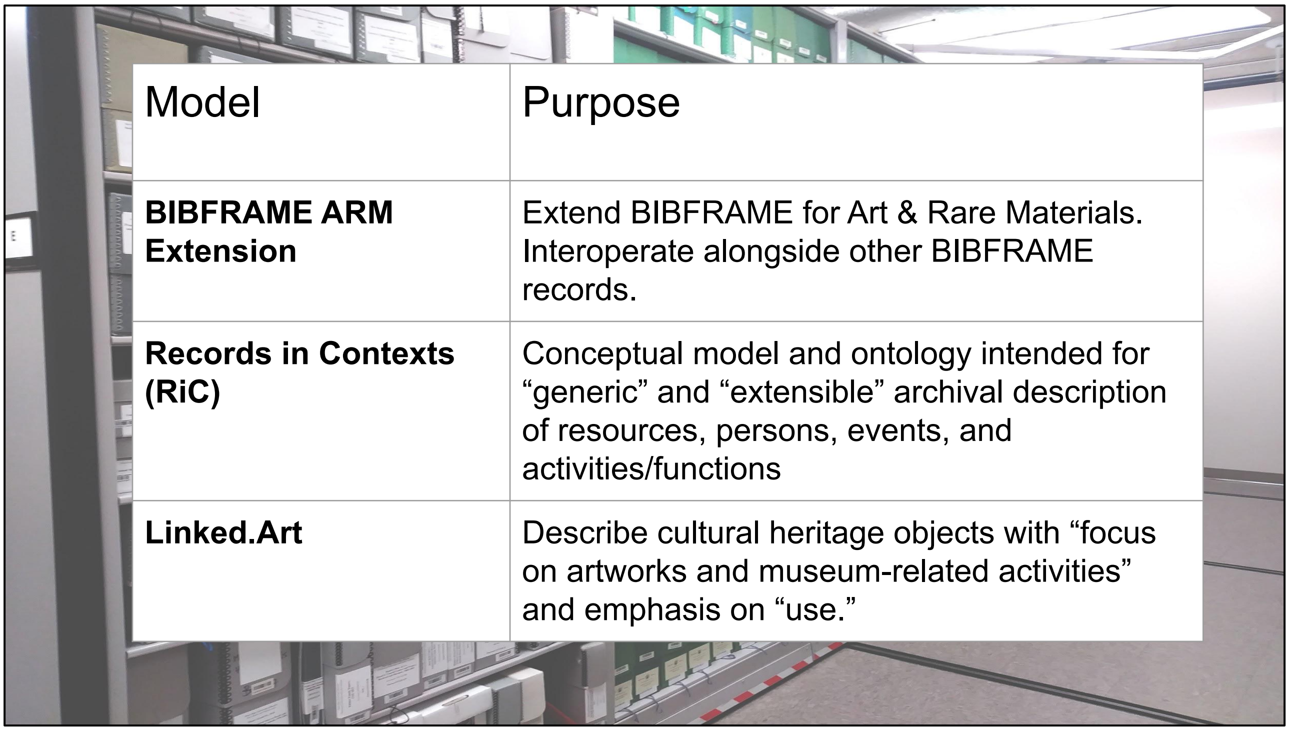


Find, Identify, Select, ***Obtain***, Explore

How do we express that retrieval object and its one-to-many relationship to descriptive aggregates in archival linked data?

Container information does not need to be modeled explicitly in our descriptive data, nor displayed in our front-facing archival websites. *But* there needs to be a way for the researcher to say “I want to consult these materials further” and for the archival staff to locate and return the containers which house those materials.

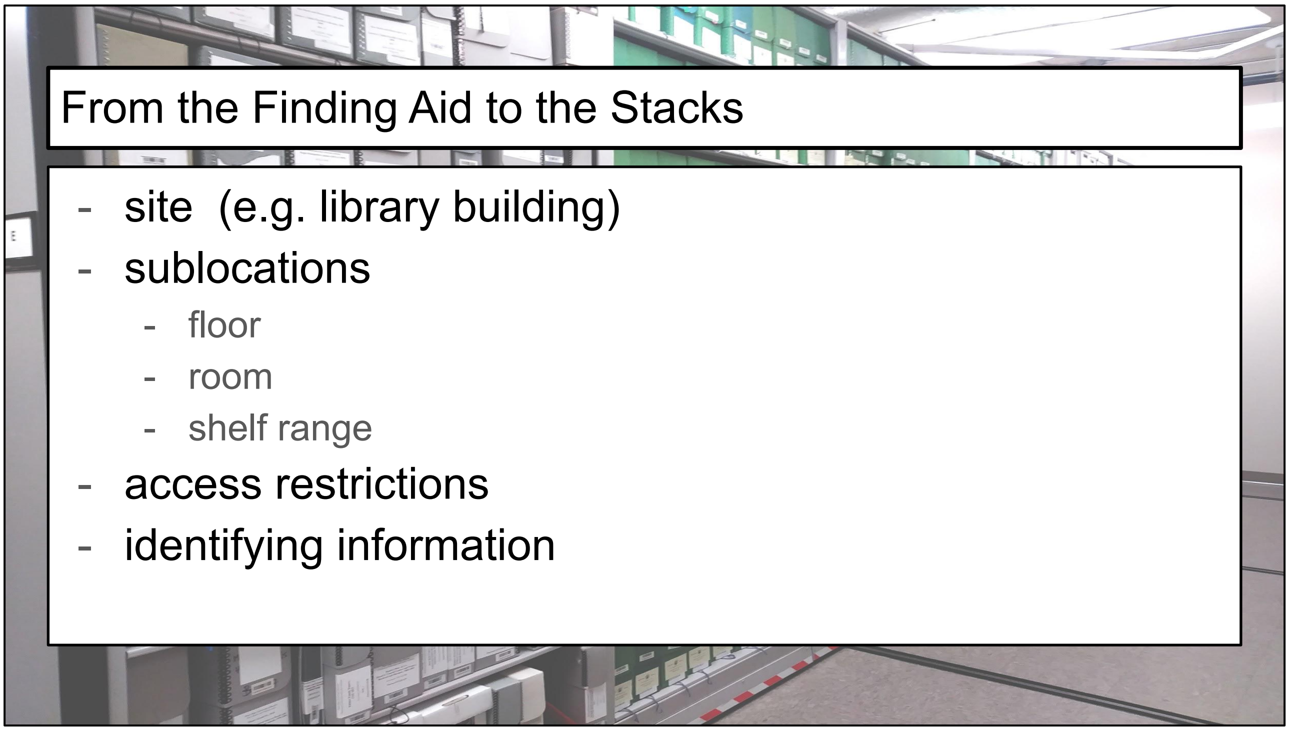
For purposes of this analysis, our focus is the *functional* rather than the technical; we are using as our framework the user tasks posited by LRM (Library Reference Model): Find, Identify, Select, **Obtain,** and Explore. Namely, how do the respective linked data models that we examined allow a researcher, having found, identified, and selected a resource of interest, to obtain that resource?



Model	Purpose
BIBFRAME ARM Extension	Extend BIBFRAME for Art & Rare Materials. Interoperate alongside other BIBFRAME records.
Records in Contexts (RiC)	Conceptual model and ontology intended for “generic” and “extensible” archival description of resources, persons, events, and activities/functions
Linked.Art	Describe cultural heritage objects with “focus on artworks and museum-related activities” and emphasis on “use.”

The three linked data models we examined are: the Arts and Rare Materials (ARM) extension to BIBFRAME; the ICA standard Records in Contexts (RiC); and the Linked.Art community model.

- ARM is a lightweight extension to BIBFRAME that came out of LD4. Its goal is to provide a mapping / transformation path for cultural heritage material to be represented as BIBFRAME, in order to bring archival description into anticipated bibliographic platforms.
- Records in Contexts consists of a Conceptual Model and an ontology specifically for archival description, replacing the suite of ISAD standards. A harmonized v1.0 is scheduled to be released in October. It’s informed by LRM. Its use is picking up in Europe (not so much in the US), with pilots by the National Archives of France, the City Archives of Amsterdam (not yet public), and the aggregator FranceArchives, among others;
- Linked.Art is heavily informed by CIDOC-CRM. We were able to access and look at data from large-scale pilots happening at the Getty and Yale’s LUX. A smaller project is happening at Van Gogh Worldwide (vangoghworldwide.org) and implementation is in progress at the Rijksmuseum Amsterdam.



From the Finding Aid to the Stacks

- site (e.g. library building)
- sublocations
  - floor
  - room
  - shelf range
- access restrictions
- identifying information

What does “obtaining” entail?

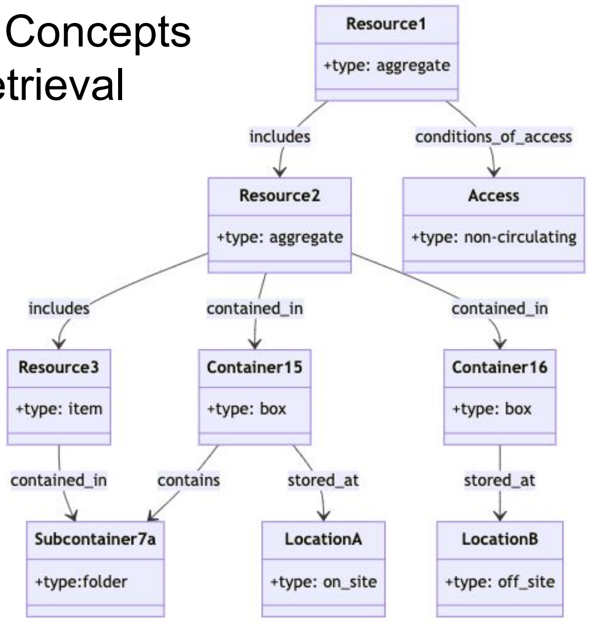
Having found, identified, and selected Edith Wharton’s invitation, it is now retrieval time. Someone needs to go get the box, a process called paging, which can be done by people or automated systems.

And just what is it that the paging agent needs to know?

Princeton, where this document resides, uses a requesting system (called Aeon) that receives requests from an online finding aid, sends them to the appropriate queue in the request database, and prints out a call slip for the human page to refer to when heading into the stacks.

Shown here is some of the information needed to guide them to the right object on the shelf–or to deny the request, in the case of restricted access.

Some Archival Concepts Playing into Retrieval



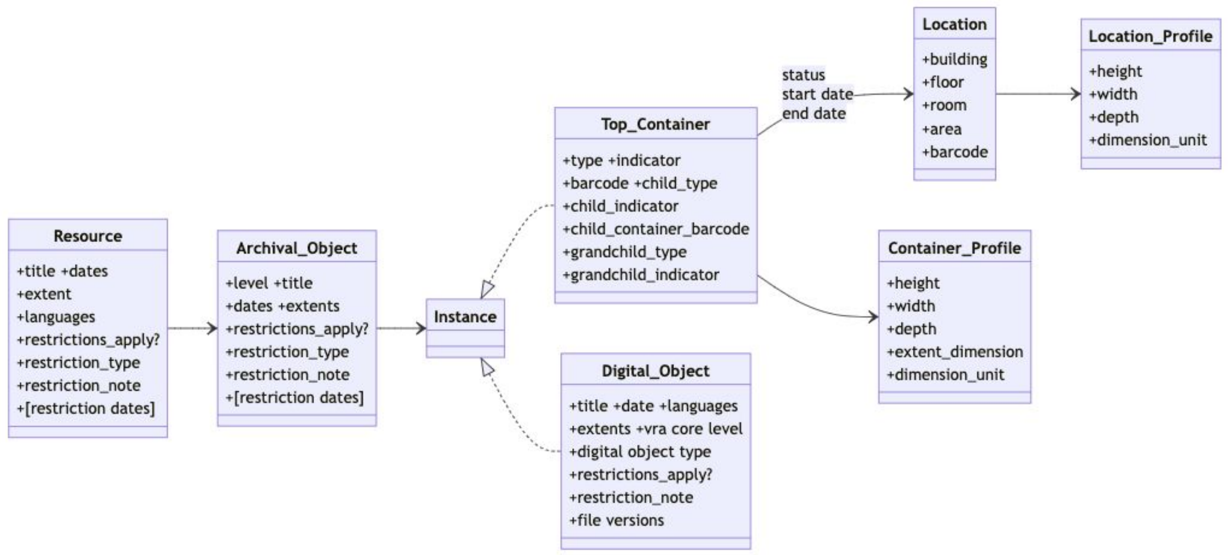
Again, containers do not need to be modeled in a descriptive ontology. Rather, the model needs to provide a docking place of sorts where a discovery system and a requesting system can communicate with one another and translate a patron's request for an intellectual resource into a request for a physical retrieval object.

What we found is that the liminal space where that handoff from one system to another occurs or should occur is often overlooked in data models, despite being vital to the purpose of archives.

This diagram illustrates the kind of "translation" that needs to happen in the contact zone between descriptive and retrieval data:

- A request for resource 3 will result in the retrieval of box 15 from location A
- A request for resource 2 (an aggregate) will result in the retrieval of two boxes, 15 and 16, from two locations, A and B, respectively
- NB neither the item (tea invite) nor the subcontainer (folder) are retrieved by themselves

Container Modeling in ArchivesSpace

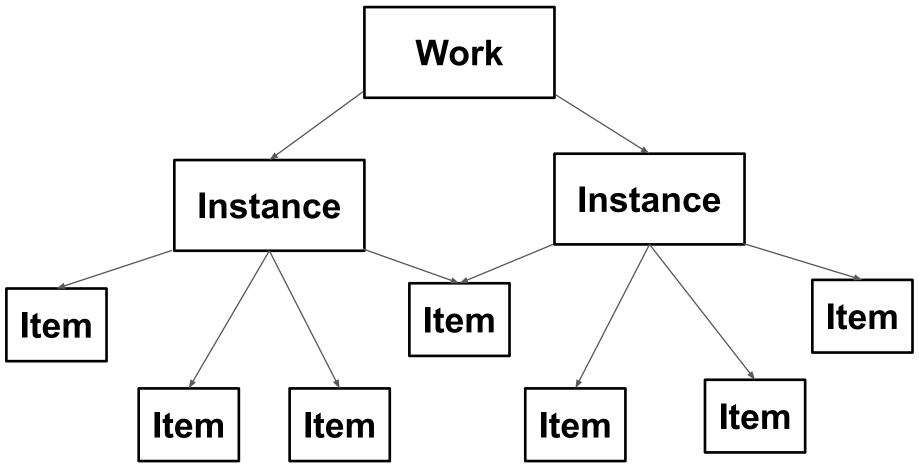


We start, counterintuitively, with ArchivesSpace, which is not linked data at all but an archival content management system widely used in the US.

The reason ArchivesSpace is an interesting case study is because it was originally released without the top_containers module shown in this diagram. The community quickly realized that ArchivesSpace needed to integrate with an end-to-end service workflow, which required additional modeling. Yale university contracted out the development of a container plug-in to the ArchivesSpace software, and that plugin was subsequently merged into the core code of the application in 2016.

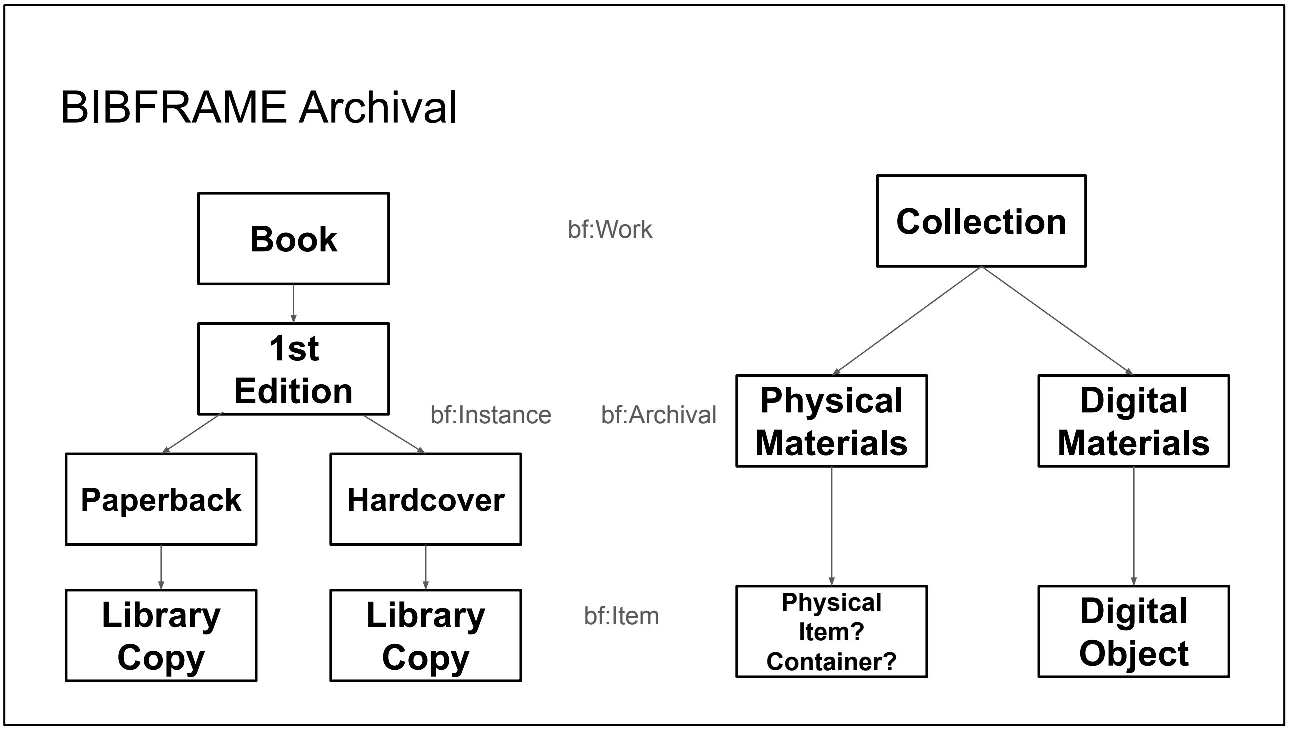
Without the top_containers module, ArchivesSpace worked perfectly fine for intellectual description: it supported discovery (find/identify). But with it, it also supports the operational needs that come with user service (select/obtain).

BIBFRAME



So the question we put to each of the three LD models is: Where is the edge, figuratively and ontologically speaking, where retrieval information might hook in and allow for a handoff of relevant data to a retrieval mechanism? Is the model extensible such that it can integrate with a retrieval-oriented model, and if so, how, concretely, does that work?

Shown here is the familiar BIBFRAME model: an intellectual work is instantiated in concrete forms, “reflecting an individual, material embodiment of a Work”, and each Instance has exemplars called Items. Intellectual aggregates can be expressed by nesting Works in other Works.



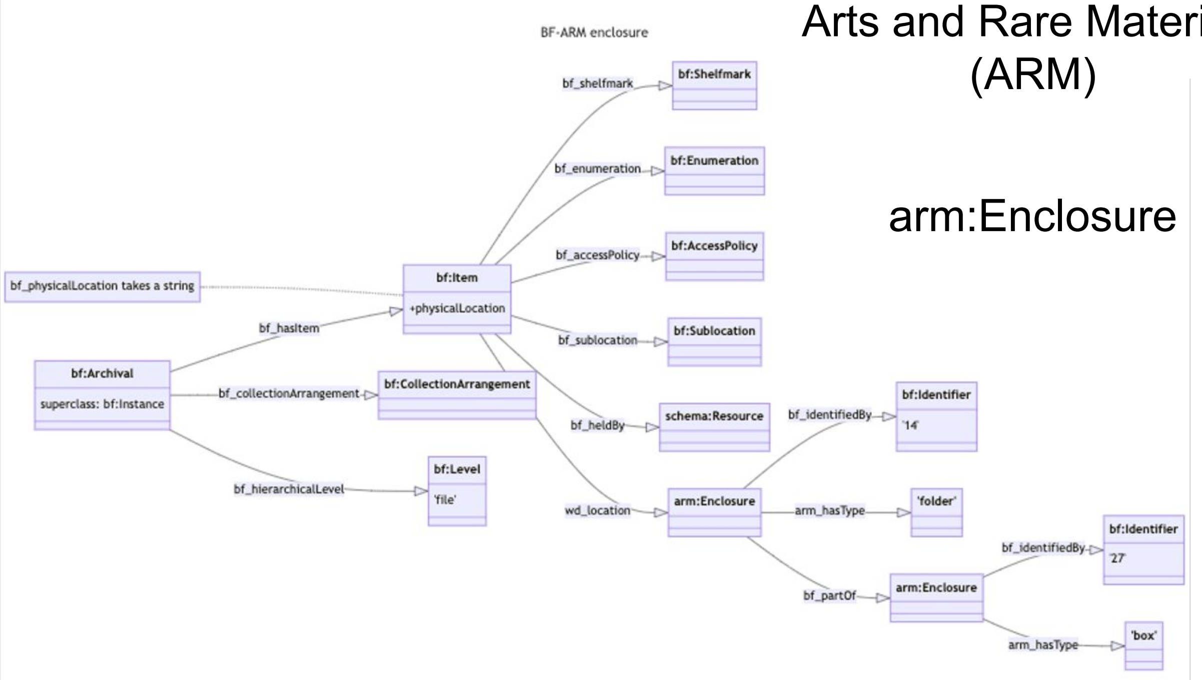
BIBFRAME provides a subclass to bf:Instance, called bf:Archival, to express "Resources organically created, accumulated, and/or used by a person, family, or organization in the course of conduct of affairs and preserved because of their continuing value."

As noted earlier, BIBFRAME is a bibliographic model, and in bibliographic models the retrieval object is typically the descriptive Item. Accordingly, all properties relevant to retrieval (access, shelfmark, enumeration, sublocation…) are on the Item level.

This remains true for the ARM extension.

Arts and Rare Materials (ARM)

arm:Enclosure



The ARM extension to BIBFRAME adds an explicit container entity, arm:Enclosure, defined as a “Covering or container to provide protection, identification, and/or keep constituent parts together.[…]”

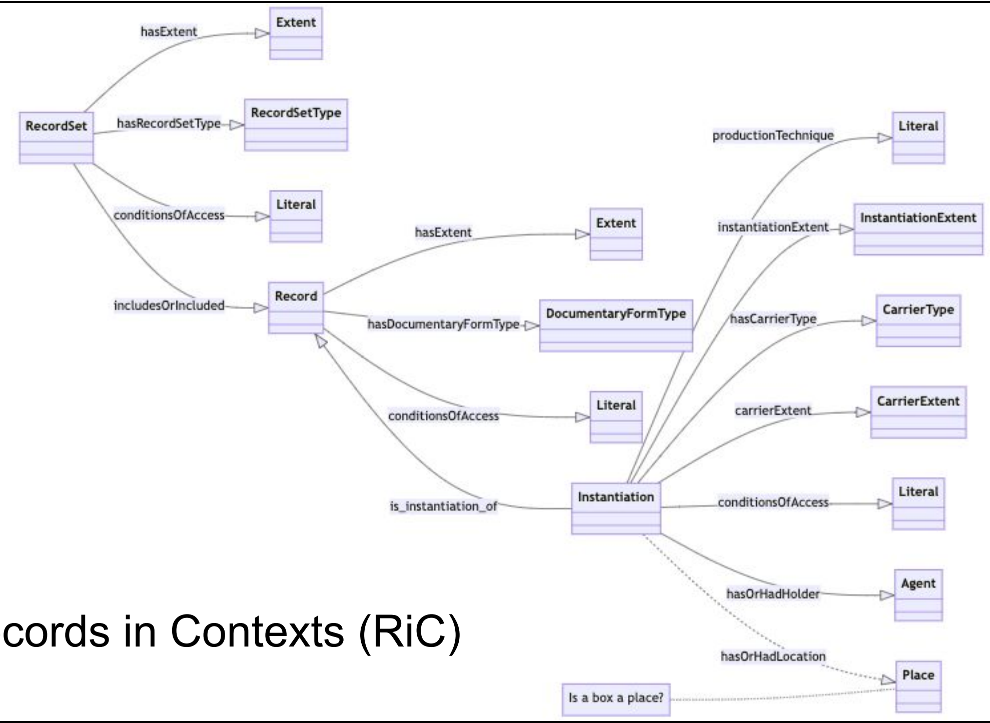
Neither BIBFRAME nor ARM have a native property that connects the arm:Enclosure with the bf:Item, but an external vocabulary may be leveraged to accomplish that. In this diagram, we’ve used a general-purpose property, wikidata:location (P276), to associate an arm:Enclosure (a folder) with a bf:Item; the folder is identified via bf:identifiedBy and has a bf:partOf that’s another arm:Enclosure (the containing box).

That makes it possible to associate a container and use it to pivot from BIBFRAME-ARM to a separate container and location management model. Since the properties relevant for retrieval and service are still associated with bf:Item, not with arm:Enclosure, it currently falls to such an external system to express the properties for the container, possibly by using an equivalent entity linked to the arm:Enclosure instance.

Further work on ARM might thus include adding the necessary properties on arm:Enclosure along with the corresponding constraints to avoid the redundancy of having the same properties on bf:Item and arm:Enclosure.

Additionally, to make ARM fully compatible with archival description, further extension of the bf:Item entity is needed such that it will allow for a compound structure capable of accommodating physical aggregates, which could be accomplished e.g. by subclassing a new arm:CompoundItem entity to bf:Item.

Records in Contexts (RiC)

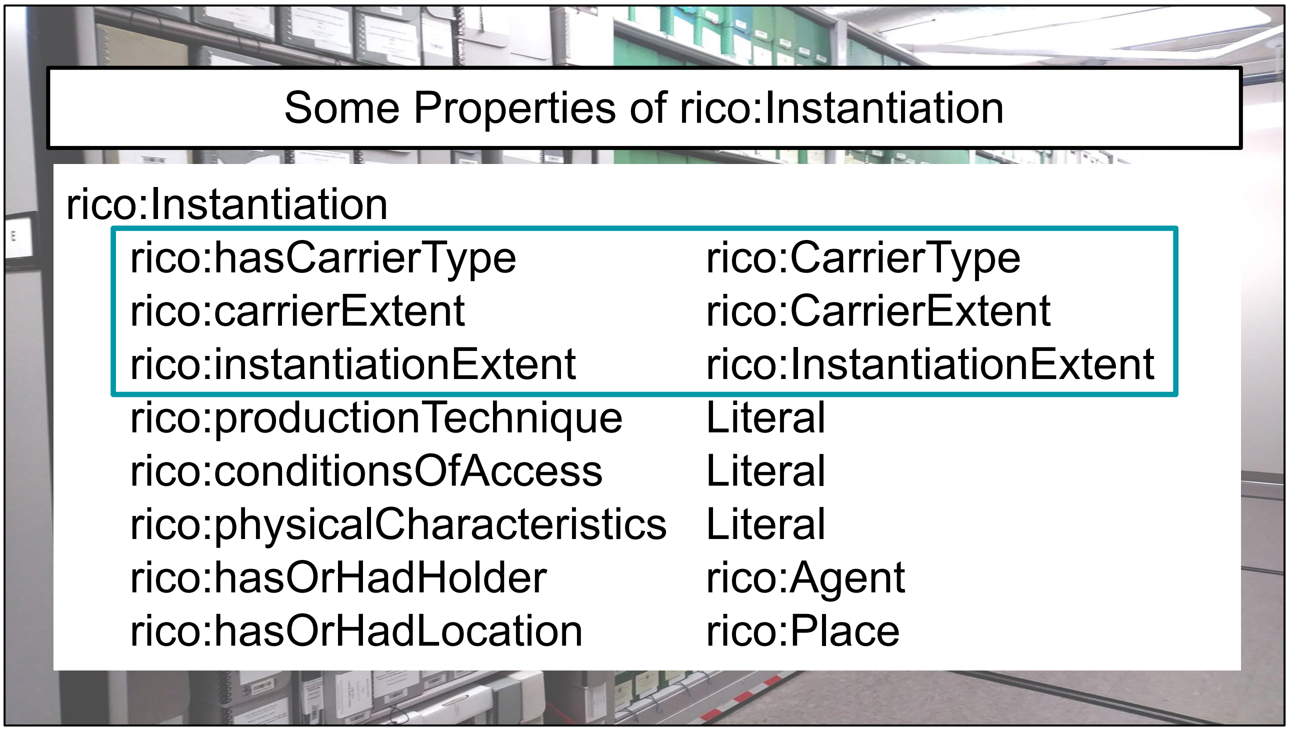


RiC does an excellent job modeling archival aggregates by introducing three subclasses of Record Resource: Record Set (for aggregates), Record (for single or compound records), and Record Part.

As shown in this (abbreviated) diagram, RiC distinguishes between the Record Resources, which are purely intellectual entities, and their physical Instantiations. rico:Instantiation may be associated with any level of a rico:RecordResource and can be chained together hierarchically using the "hasComponent" property (not shown here).

RiC and RiC-O have no concept of a container and implicitly emphasize intellectual over physical control. They therefore seem best positioned to support secondary discovery sites like aggregating services (e.g. FranceArchive), as they privilege discoverability and, subsequent to discovery, will redirect researchers to a primary finding aid from where any specific location and retrieval information may be negotiated.

With no Container entity defined, RiC and RiC-O are thus less suited to replace the primary finding aid outright, since a finding aid typically does inform the user where and how to obtain a resource. Some early implementations trying to express containers natively within RiC/RiC-O anyway have, for the moment, worked around this impediment by declaring containers instances of rico:Place (which can be associated natively with a rico:Instantiation).



Some Properties of rico:Instantiation

rico:Instantiation

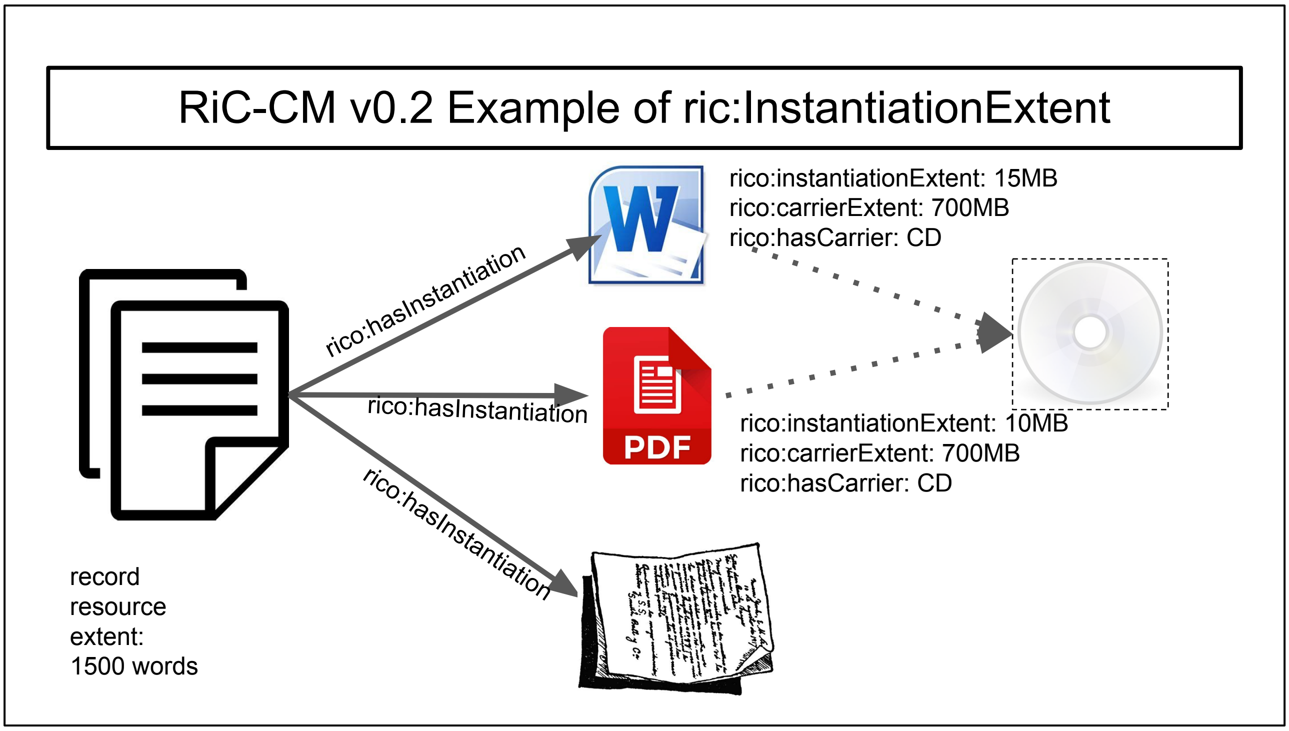
rico:hasCarrierType	rico:CarrierType
rico:carrierExtent	rico:CarrierExtent
rico:instantiationExtent	rico:InstantiationExtent
rico:productionTechnique	Literal
rico:conditionsOfAccess	Literal
rico:physicalCharacteristics	Literal
rico:hasOrHadHolder	rico:Agent
rico:hasOrHadLocation	rico:Place

The deeper surprise, though, may come with the conceptualization of the ric:Instantiation entity itself.

Instantiation is described as: “The inscription of information made by an Agent on a physical carrier in any persistent, recoverable form[...]” and it is also what’s held by an Agent, via the hasOrHadHolder property, or is located at a Place, via the hasOrHadLocation property.

That suggests that rico:Instantiation is an entity similar to a bf:Item.

However, that doesn’t fully mesh with the fact that both carrierExtent *and* instantiationExtent are properties of Instantiation.


RiC-CM v0.2 Example of ric:InstantiationExtent
rico:hasInstantiation
rico:hasInstantiation
rico:hasInstantiation
rico:instantiationExtent: 15MB
rico:carrierExtent: 700MB
rico:hasCarrier: CD
rico:instantiationExtent: 10MB
rico:carrierExtent: 700MB
rico:hasCarrier: CD
PDF
record
resource
extent:
1500 words

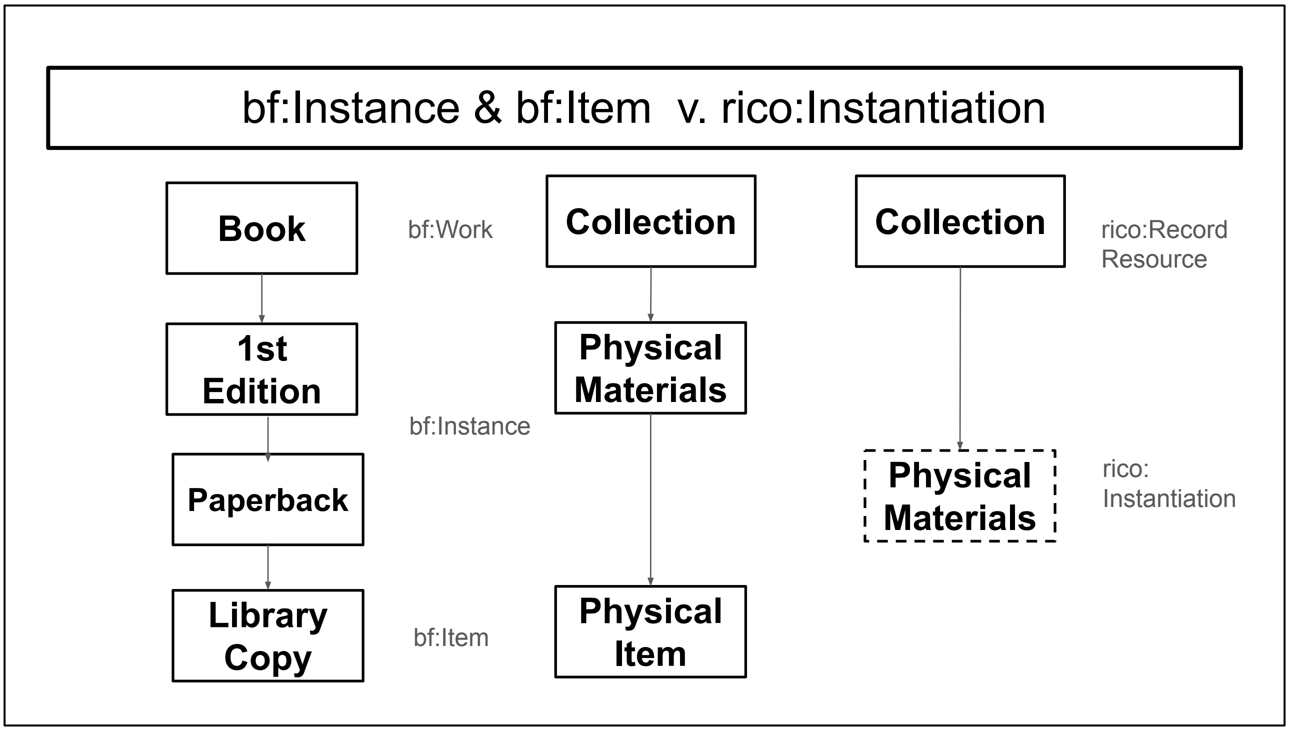
In fact, in examining the meaning of those two properties on rico:Instantiation more closely, it appears that rico:Instantiation also shares some characteristics with bf:Instance and even lrm:Manifestation, which, for comparison, is defined as:
“A set of all carriers that are assumed to share the same characteristics as to intellectual or artistic content and aspects of physical form[...]”.

That is because, according to the RiC Conceptual Model, each time the same content is instantiated with a different set of properties (e.g. production technique), that constitutes a distinct Instantiation: for example, “a record printed and saved at the same time as both DOCX and PDF/A would have three concurrent instantiations.” (p.25)

The CM further explains the difference between *carrier extent* and *instantiation extent* by ways of an example that reinforces that same point:

“For example, a CD with a storage capacity of 700 MB (carrier extent) might hold a record of 1500 words (record resource extent) represented in two versions, one a Word document with an instantiation extent of 3 KB and the other a PDF file with an instantiation extent of 5 KB.” (p.43, 53, 58)

That no longer describes an Item but, rather, a Manifestation along the lines of the bf: Instance or the concept of the FRBR/FRBR-OO/LRM Manifestation.



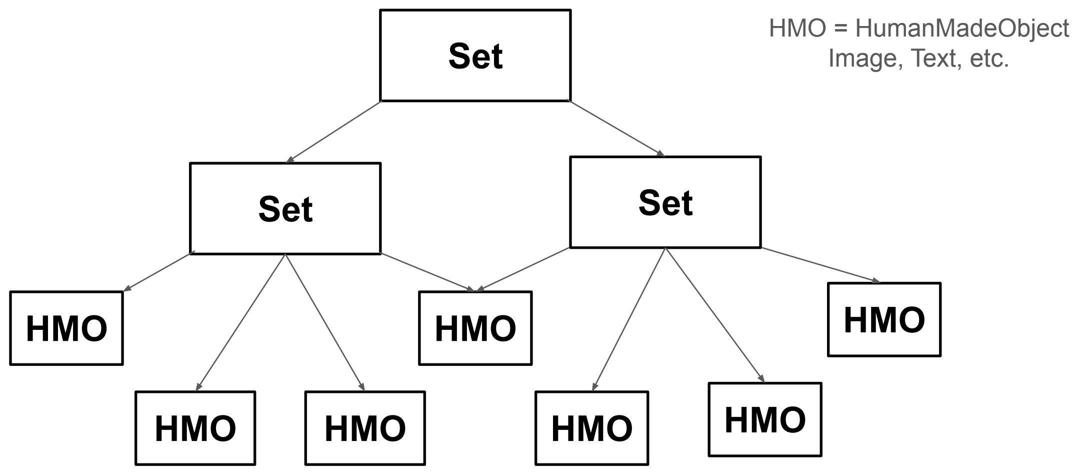
This suggests that ric:Instantiation may be undermodeled in that it assumes that in the archival domain, the manifestation and its exemplar are *physically one*, and can be collapsed into a single descriptive entity.

It makes rico:Instantiation an inherently unstable entity: like Schrödinger's cat, the rico:Instantiation is a bf:Instance *and* a bf:Item at the same time, yet will behave either like the one or like the other depending on when you look. (Likewise for a container modeled as a rico:Place.)

rico:Instantiation is therefore the entity that we might reasonably expect to be where a Container entity would hook in: for example analogous to ARM, i.e. via an external property and an additional, externally defined, Container entity that is in a same-as relationship to the arm:Enclosure.

But unlike BIBFRAME-ARM, where the bf:Item is a discrete concept distinct from the bf:Instance (denoting a single exemplar out of many exemplars that make up one bf:Instance), the rico:Instantiation is ambiguous. This would indicate that any relationships it enters, be it to entities defined within RiC or outside it (like an external Container entity), are likely to yield conceptually inconsistent query results.

Linked.Art

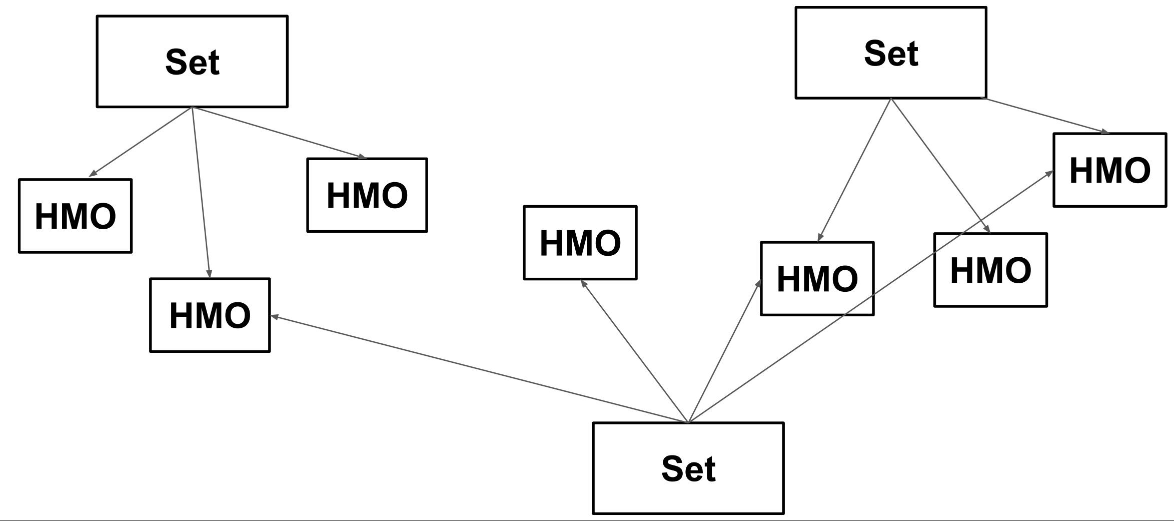


When describing materials, Linked.Art uses the CIDOC-CRM concept Human-Made Objects, which can be organized into Sets. It provides some specific fields for the subtype of object, e.g. a Digital Object, and core generic fields that can then be defined through "classified as" statements which describe the role they're playing in the description.

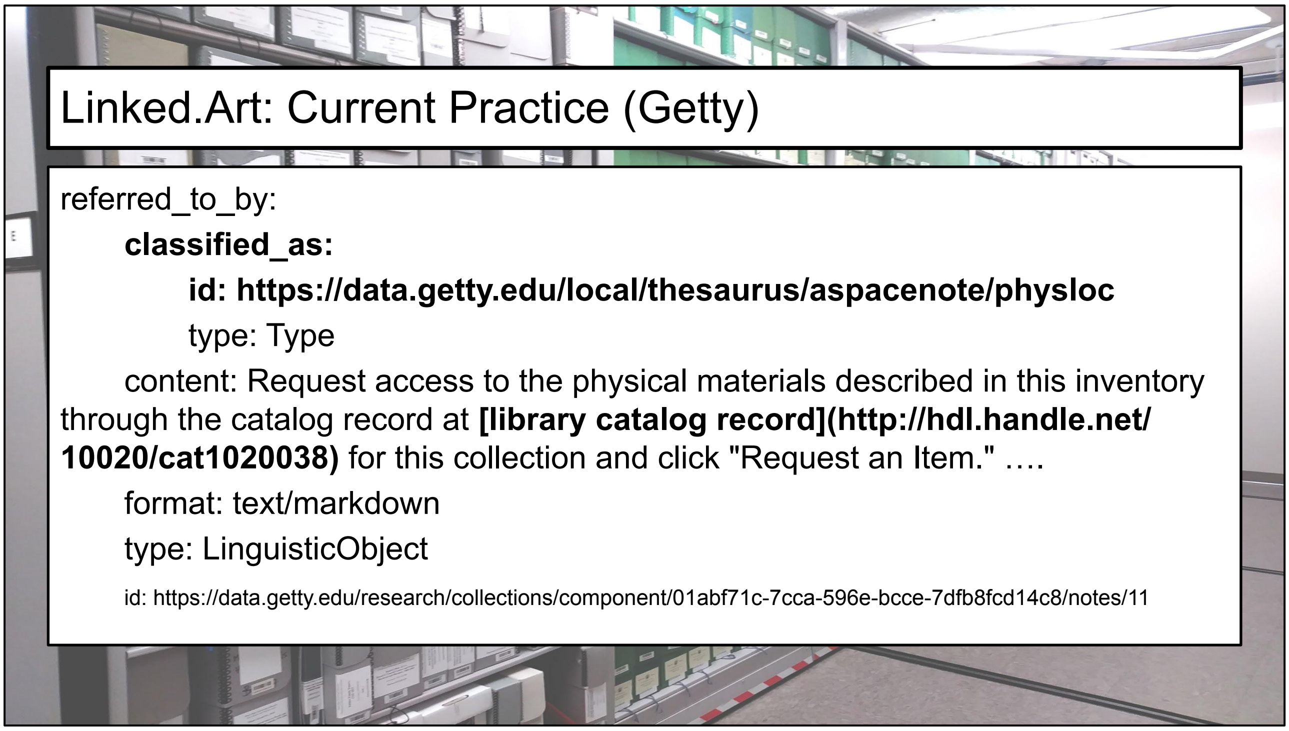
When it comes to archives, it should be noted that HumanMadeObjects don't have to correspond to a single letter or sheet of paper, just the lowest level of description (which could be an aggregate).

This slide shows a fairly traditional concept of description, but…

Linked.Art



…in linked data, there can be multiple contexts.



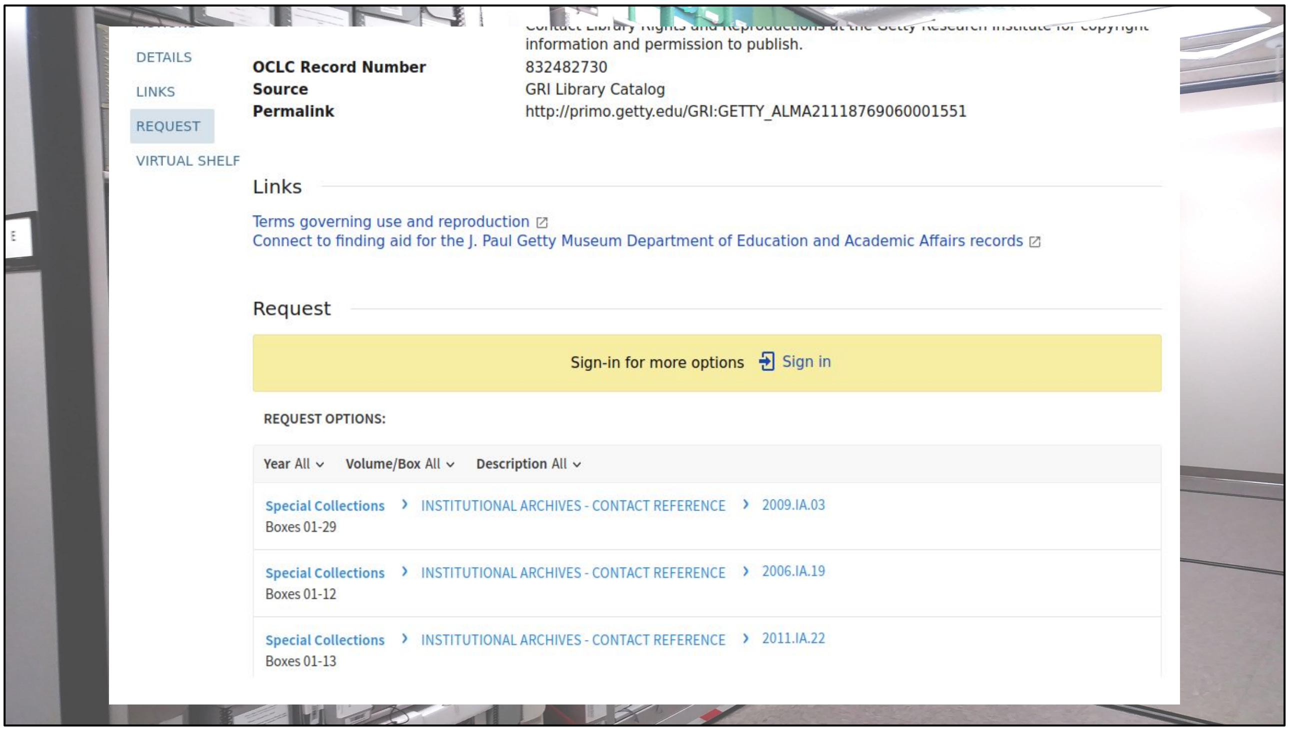
Linked.Art: Current Practice (Getty)

```
referred_to_by:
      classified_as:
            id: https://data.getty.edu/local/thesaurus/aspacenote/physloc
            type: Type
      content: Request access to the physical materials described in this inventory through the catalog record at [library catalog record](http://hdl.handle.net/10020/cat1020038) for this collection and click "Request an Item." ….
      format: text/markdown
      type: LinguisticObject
      id: https://data.getty.edu/research/collections/component/01abf71c-7cca-596e-bcce-7dfb8fcd14c8/notes/11
```

For Linked.Art, we were able to look at production data, including a retrieval path, from the Getty and Yale's LUX. The folks designing both of these systems had to solve the problem we're looking at here, how to get the researcher to the resources they want. It is worth noting that both systems are still experimental and evolving, which means that what's shown here is how they're making it work right now.

This slide shows some pseudoJSON to illustrate the Getty's modeling. As mentioned, Linked.Art has generic structures which can be typed - so what is shown here is essentially a Linguistic Object (a note) formatted in Markdown, classified as a locally-defined type: a Physical Location ASpace Note.

The link to the library catalog record is embedded in the "content" field, so if the researcher is interested in this resource, they'll follow the link over to Primo, where…



DETAILS
LINKS
REQUEST
VIRTUAL SHELF

information and permission to publish.

OCLC Record Number	832482730
Source	GRI Library Catalog
Permalink	http://primo.getty.edu/GRI:GETTY_ALMA21118769060001551

Links

Terms governing use and reproduction
Connect to finding aid for the J. Paul Getty Museum Department of Education and Academic Affairs records

Request

Sign-in for more options Sign in

REQUEST OPTIONS:

Year All Volume/Box All Description All

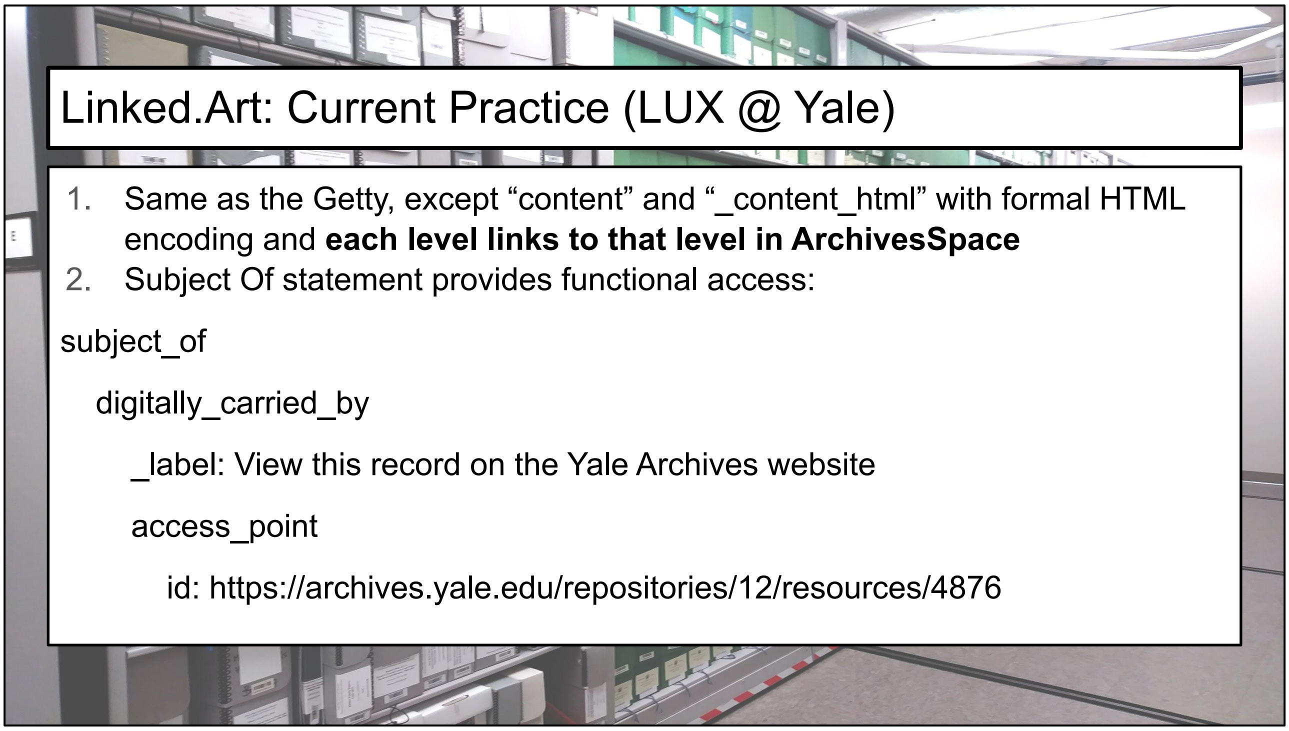
Special Collections › INSTITUTIONAL ARCHIVES - CONTACT REFERENCE › 2009.IA.03
Boxes 01-29

Special Collections › INSTITUTIONAL ARCHIVES - CONTACT REFERENCE › 2006.IA.19
Boxes 01-12

Special Collections › INSTITUTIONAL ARCHIVES - CONTACT REFERENCE › 2011.IA.22
Boxes 01-13

…they can select boxes from this request list.

For the Getty system, the link remains the same, no matter what level of description you’re on.



Linked.Art: Current Practice (LUX @ Yale)

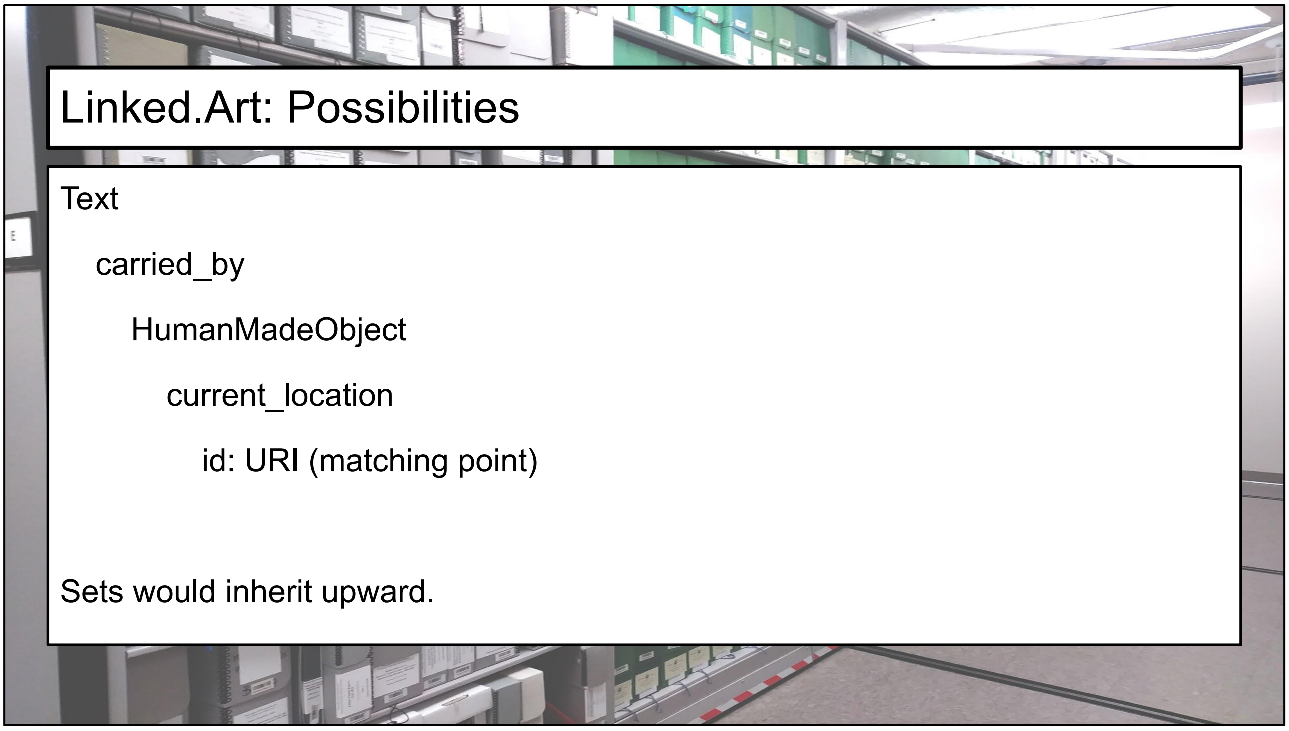
1. Same as the Getty, except “content” and “_content_html” with formal HTML encoding and **each level links to that level in ArchivesSpace**
2. Subject Of statement provides functional access:

```
subject_of
   digitally_carried_by
      _label: View this record on the Yale Archives website
      access_point
         id: https://archives.yale.edu/repositories/12/resources/4876
```

LUX does something similar. Unlike in the Getty, the URLs in LUX vary by encoding and take the researcher to their respective representations at the same *level of description* in Yale’s ArchivesSpace. Request links are then available from the ArchivesSpace sidebar once the researcher is on a low enough level of description.

In addition, LUX has a subject_of property that associates a unit of description (e.g. a series, subseries, item etc.) to a digital object (the ArchivesSpace webpage). It includes an “access_point” entity with an identifier “id.” Compared to the Getty solution of putting a link in markdown or HTML, this is a more structured way of encoding this kind of information.

This construct is not intended as a formal access or use statement; the modeling would likely look different for actionable access restrictions.



Linked.Art: Possibilities

```
Text
    carried_by
        HumanMadeObject
            current_location
                id: URI (matching point)
```

Sets would inherit upward.

The access_point / ID pairing allows for expressing a digital surrogate or original digital object.

Access_point is defined as “A URL from which the digital object is able to be retrieved” and is only available on DigitalObject. A Text object can take a carried_by property, which refers to a HumanMadeObject, which in turn can have a current location. The Current Location’s ID needs to be a URI, but this URI could be paired with a box ID and barcode in the retrieval system (or even *be* a box barcode or a parameterized URL, though this would likely not be best practice.)

At the Set level, one might collect and display an aggregation of all the current locations of Things in the Set.

Where Does a Beautiful Ontology Meet Actual Stuff?

A researcher walks into an archives. They request a compound record. They receive a box, or several boxes.

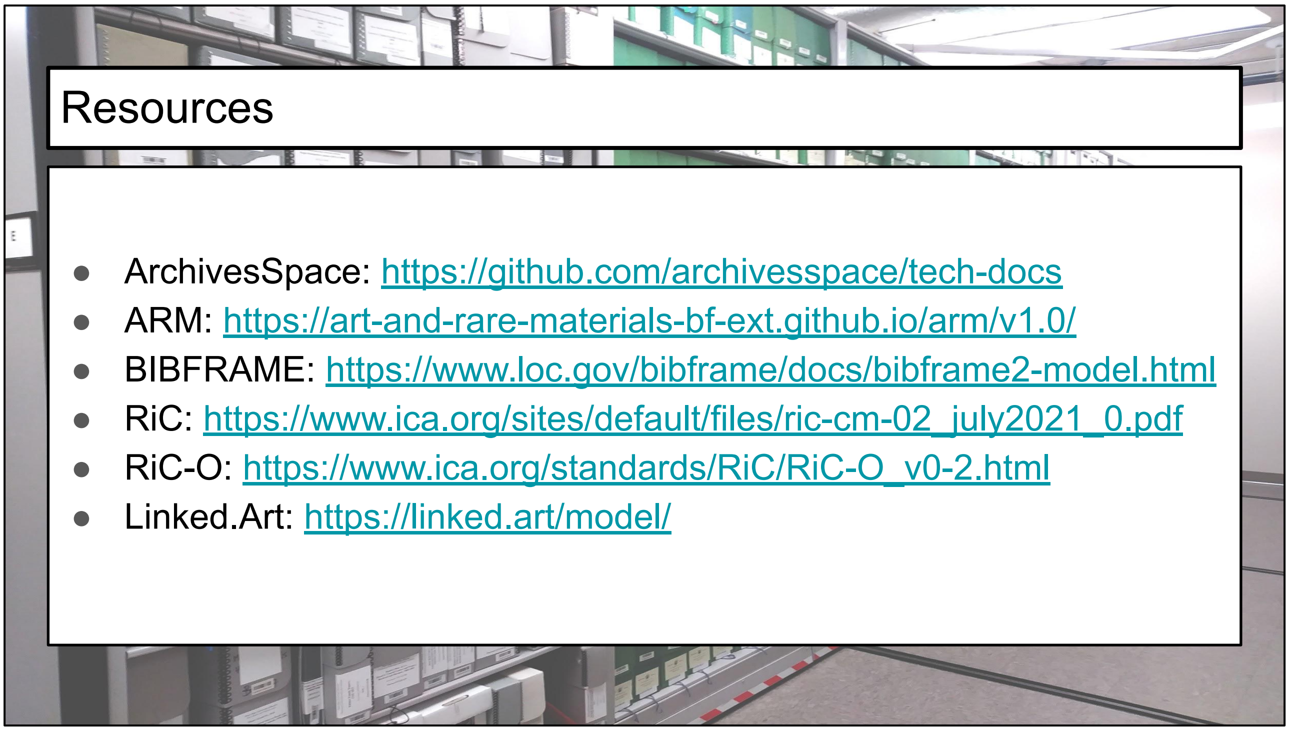
Someone, somewhere, whether human or machine, has to figure out how the requested resource maps onto the retrieval unit. That is the core function of an Archive.

Our analysis of three emerging archival linked data standards shows that they either, like RiC/RiC-O and Linked.Art, privilege intellectual description over physical description, which precludes seamless integration with a request system; or, like ARM, privilege item-level over aggregate-level description, which precludes aggregate-level requesting.

Workarounds are possible, as Linked.Art and RiC implementations have shown; they include modeling a container as an instance of Place or linking out to the primary finding aid for the last mile to the retrieval handoff. Boutique workarounds like these are serviceable–but they require each repository to expend resources on hacking its way to core functionality, projects that are rarely every shareable.

Our data models should be supporting Obtaining as a primary objective, not an afterthought. Rather than create bespoke one-off solutions on the implementation, rather than the modeling, layer that will likely age and scale poorly, we need to position description for operational needs. Our models don't need to express Containers explicitly, but they need to take into account that retrieval is the endgame, and that therefore any descriptive model needs an edge onto which Containers can be attached.

No matter how detailed and informative our intellectual the description is, we don’t believe that a model can be sustainably implemented as a primary pathway into our archives until a researcher can complete the journey.

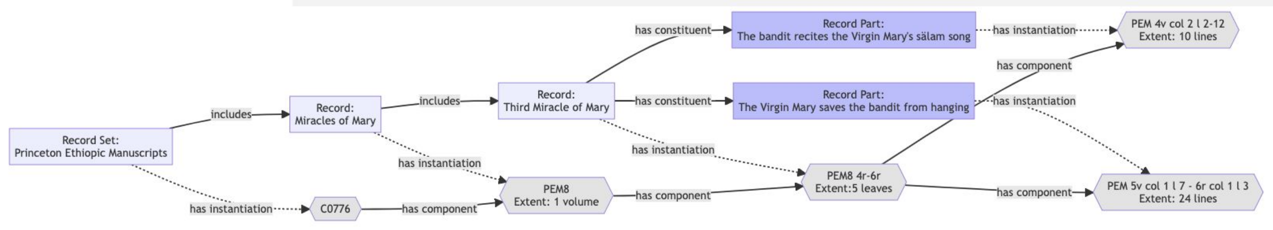


Resources

- ArchivesSpace: https://github.com/archivesspace/tech-docs
- ARM: https://art-and-rare-materials-bf-ext.github.io/arm/v1.0/
- BIBFRAME: https://www.loc.gov/bibframe/docs/bibframe2-model.html
- RiC: https://www.ica.org/sites/default/files/ric-cm-02_july2021_0.pdf
- RiC-O: https://www.ica.org/standards/RiC/RiC-O_v0-2.html
- Linked.Art: https://linked.art/model/

Questions?

Contact us
Ruth: rkt6@psu.edu
Regine: heberlei@princeton.edu



Illustrating “rico:Instantiation at any level”