This is an Accepted Manuscript of an article published by Taylor & Francis in Journal of Archival Organization on 19 Jan 2018, available online: http://www.tandfonline.com/ DOI: 10.1080/15332748.2017.140072

Opportunities for Encoding EAD for Linked Data Extraction and Publication

Introduction

In 2015, the Technical Subcommittee for Encoded Archival Description of the Society of American Archivists released the third revision of the Encoded Archival Description (EAD) standard. This version, commonly referred to as EAD3[1], supersedes the second version released in 2002 (commonly known as EAD2002), although lack of tooling has limited adoption thus far.[2] While EAD3 is a hierarchical, XML-based metadata encoding standard[3] used to create archival finding aids, not a linked data standard, it includes elements and attributes which provide opportunities for archivists to encode finding aids from which linked data may easily be extracted.

Through sharing archival description as linked data, archivists may connect the collections they hold to the semantic web. While Google, for example, does not understand what a <persname> means in <origination> and may not accurately parse a finding aid transformed into HTML, it understands that when a person is described as a "creator" using the schema.org ontology[4], that person was responsible for the creation of materials. The archival world is actively working to improve these interchanges. At the end of September 2017, a recommendation was submitted to schema.org for an extension of their fields to differentiate archives from local businesses and indicate that materials described are archival or a single work in a broader archival collection.[5] The new ArchivesSpace[6] Public User Interface includes automatic extraction and publication of certain fields as linked data in schema.org. An

unsponsored Archives and Linked Data Interest Group has developed a recommended archival linked data mapping between schema.org, the ISAD(G) standard, and Describing Archives: a Content Standard (DACS). The group intends to pursue additional avenues of mapping for exchange and use of archival description beyond their initial and pragmatic choice of schema.org.[7] And the ICA-sponsored Experts Group on Archival Description intends to express their new combination of archival descriptive standards, Records in Context, as an Web Ontology Language (OWL) ontology.[8]

This paper is an overview of aspects of EAD3 which allow for one to create linked data-ready archival description. It explores the ways in which one could encode the URIs and other descriptive components required for linked data before EAD3 and how EAD3 refines existing features and adds the EAC-CPF concept of relations. It then provides some context for why one may wish to bring one's archival description into the semantic web and a brief overview of the nature of RDF statements. After sections going over the technical aspects of the two main ways of encoding intentional linked data-ready archival description, it briefly treats other types of data which one may wish to extract from finding aids and express as RDF before concluding.

Note on syntax: throughout the article, XML elements will be expressed within angle brackets, e.g. <ead>. XML attributes will be represented with an @ symbol in front of them, e.g. @identifier.

EAD3 Cultivates Existing Potential Within EAD

From its inception, EAD has contained within it elements and attributes which could be used to encode URIs for use as linked data. Its earliest form did not make this readily apparent. EAD 1.0, released in 1998, patterned itself on the TEI document markup standard in its header and tag naming conventions.[9] Elements such as <runner>[10] and the entire <frontmatter>[11] section reflect its self-perception as a way to exactly replicate the print finding aid in an XML form. But, alongside those, it included such elements as <subject> and <persname> in which it

provided granular methods for encoding not just the text of an authorized heading or name but identifiers (@authfilenumber), sources (@source), and, in the cases of names, a description of their @role relative to the collection. Although early adopters may not have had linked data in mind when they used them, if they used these attributes when referencing authorities, they laid the foundation for their successors to transform and reuse their data as linked data. Indeed, as the next section demonstrates, it is possible to encode for Linked Data within EAD2002 (or EAD 1.0). Although the EAD2002 tag library still defines attributes like @authfilenumber without references to URIs or linked data, the author of the entry for <persname> writes "The AUTHFILENUMBER attribute can be used to identify a *link* to an authority file record that has more information about the name or cross references for alternative forms of the name and related names."[12]

EAD3, then, has built from a strong foundation. It has borrowed again from a cultural heritage encoding standard, this time EAC-CPF with its emphases on relationships and options for further granularity of description. Besides adding new elements and conceptions of how relationship might look within a finding aid, the subcommittee tasked with updating EAD renamed attributes within the standard to align with names and practices within the linked data and MARC/RDA descriptive world, making existing parallels explicit to the reader through these alignments. And when writing the accompanying tag library, the subcommittee pointed the finding aid author toward URIs and linked data, exposing the possibilities within the standard. This last is crucial, as documentation influences data practice for those who cannot read technical specifications. What is left out may be perceived as not allowed, particularly by those who rely on documentation because they stumble over the full meaning of Document Type Declarations. Despite the statement about @authfilenumber shared above, the official definition: "A number that identifies the authority file record for an access term drawn from that authority file"[13] would not seem to allow one to use a URI. Yet URIs are technically valid and even encouraged within ArchivesSpace, the most recent system to emerge for the generation of

EAD2002. So although EAD3 presents further opportunities for encoding data which may be extracted and used as linked data, and an entirely new concept of <relation>, a migration to EAD3 is not necessary for the repository whose workers wish to use its finding aids as sites in which they may encode and from which they may extract linked data. Readers will encounter methods which can be used in EAD2002 as well.

What, then, is the scope of this paper? This paper introduces the reader to minimal concepts of linked data, pointing them at other resources which may provide more robust definitions. It then moves iteratively through EAD's support for linked data from EAD 1.0 onward, demonstrating how repositories using EAD2002 may transition to linked data-ready encoding, or may be unaware that their local practices already allow for the extraction of linked data from their records. It then explains how EAD3 refines what has come before, how it solves some problems with existed, and how its newer naming practices bring it more in line with other standards. After treating this avenue of encoding within EAD for extraction as linked data, it moves on to a second and entirely new method <relations> and how finding aid authors may use this to dramatically re-imagine the finding aid, both as a site of textual description and of encoding.

Linked Data and Archival Relationships

"Linked data uses the resource description framework[14] and is expressed as three-part statements called triples, each triple consisting of a subject (what the triple is about), a predicate (describing the relationship of the subject to its object), and the object (describing an attribute of the subject or identifying the subject of another triple to which it is related)."[15] It "allows computer systems to share information on the Semantic Web, thus *enabling data from different sources to be connected and queried*."[16] The subject and predicate must be a URI and the third will be so whenever possible. These triples consisting of three URIs are known as "five-star"

linked data.[17] To attempt a translation of a five-star triple into English, one might have the following:

```
<URI representing a collection> <the URI to the MARC relator definition
of Creator> <A URI to Walter Mosley's Authority Record> .
```

The statement translates "this collection has the MARC relator-defined creator Walter Mosley" or "Walter Mosley, as described by this authority record, created the materials in this collection." One could simply use the string literal[18] or authorized text of his name, "Mosley, Walter" and make the statement:

```
<URI representing a collection> <the URI to the MARC relator definition
of Creator> "Mosley, Walter" .
```

This would be considered "four-star."[19] Although a string literal such as "Mosley, Walter" or "Walter Mosley," has some value, not including URIs wherever possible reduces one's decision to use RDF to a decision to encode one's textual description as a series of three-part statements--not wrong, but not productive.[20] A section near the end of this paper will treat some of the ways in which meaningful text can be extracted from EAD3 as RDF. The majority of the paper, however, will focus on URIs as objects within RDF, as this is what allows full semantic querying to occur.

Triples may be encoded in a variety of languages, from XML (RDF/XML) to JSON (JSON-LD). Where possible, examples will be presented as Turtle,[21] a "terse" method of expressing RDF which lessens the amount of page space used and reinforces the idea of a triple as subject, predicate, and object. In Turtle, prefixes are defined to represent segments of URIs and used as substitutes for that part of the URI. For example, one might define:

```
@prefix lcsh: <http://id.loc.gov/authorities/subjects/> .
```

After making such a declaration, one could encode the URI <http://id.loc.gov/authorities/subjects/sh85036085> as lcsh:sh85036085. This practice is similar to the association of an XML namespace with a URI at the beginning of an XML document,

except that URI of an XML namespace refers to the documentation or accepted identifier for that namespace, whereas the prefix of a triple as Turtle stands in for a URI *segment* which should be joined with the suffix to create the full URI.

If dc: then represents the URI <http://purl.org/dc/terms/> and local: represents the URI prefix of a hypothetical local repository <https://local.org/>, one might assert that:

```
local:10483    dc:subject    lcsh:sh85036085 .
```

Translated: the local resource identified as https://local.org/10483 has the subject (as specified in the DC Terms ontology, not another ontology which may conceive of "subject" differently) "Death" as defined by the Library of Congress's subject headings. This statement links out from your repository to reference external data, data which may be similarly referenced by a wide variety of repositories.

Functionally, LCSH and established name authorities do this already within the limited sphere of LAM. If one aggregated their data, as libraries already do in WorldCat and consortia, the controlled string literal "Mosley, Walter" would mean the same person for every record. But what if we wanted to extend beyond the realm of record creators who use LC's name authorities, or work creators who have the literary warrant to earn them? How can we be sure that other people mean Walter Mosley the author and not someone else who shares the same name?

Linked data not only allows one to mix and match predicate sources and authorities[22] within a single record, it allows one to declare or infer connections between these things, as well as between works. We may make assertions that our things are the same as others' things and others may make similar assertions in return. For example, LCNAF's linked data authority record for Walter Mosley asserts the following triple:

```
lcnaf:n88221921 skos:exactMatch viaf:4979209
```

Translated, his representation on LCNAF as "n88221921" is exactly the same and has an inverse relationship to his representation on VIAF as "4979209."

Then on DBPedia page for Walter Mosley asserts that it its record is owl:sameAs each of the following records for Walter Mosley.

```
viaf:4979209
wikidata:Walter Mosley
dbpedia-de:Walter Mosley
dbpedia-es:Walter Mosley
dbpedia-fr:Walter Mosley
dbpedia-it:Walter Mosley
dbpedia-ja:Walter Mosley
dbpedia-pl:Walter Mosley
dbpedia-wikidata:Walter Mosley
freebase:Walter Mosley
nyt:Walter Mosley
yago-res:Walter Mosley
http://d-nb.info/gnd/115769234
```

Note that VIAF tops the DBPedia list. Through inferencing based on that VIAF URI and its skos:exactMatch, one may assert that Walter Mosley as lcnaf:n88221921 is an exact match/same as (based on the definitions from owl and skos) all the representations of him referenced above. Therefore, if an archival institution holding Mosley's work were to make the very basic claim that its collection

```
local:39580 dc:creator lcnaf:n88221921
```

one may infer that Walter Mosley as represented on all these sites is the creator of the materials.[23] Such machine inferencing is the querying spoken of earlier.

Many authorities which archives already use to provide authorized names, headings, and the like for their description have been published as linked data. The Library of Congress Linked Data service provides a vast array of authorities, including its names, subjects, thesaurus of graphic materials, genres and forms, etc. at id.loc.gov. As will be shown below, for

many of these, there is already a relationship between the authority file number and the full URI for the same term published as linked data. Those familiar with MARC relators will be able use those as predicates, thanks again to the Library of Congress's linked data service. Although no linked data archival ontology currently exists, others used in the domain include Dublin Core and Schema.org. For more on the practical aspects of embarking on the use of linked data--consideration of linked data vocabularies, predicate mapping, and suggestions for the programmatic extraction and publication of linked data--readers may wish to consult Gracy,[24] Arnold[25] along with Salesky,[26] Rubinstein,[27] Jones and Seikel,[28] as well as numerous online tutorials on the subject.

EAD3's Methods for Encoding Linked Data

EAD3 offers two methods for encoding relationships such as that between Mosley and materials which he created (and between those materials and other people, subjects, collections, etc.) in a way that may be extracted as linked data. The first is an augmentation of existing access point elements and guidance about their use more suited to linked data. Access point elements in EAD3 are <corpname>, <famname>, <function>, <genreform>, <geogname>, <name>, <occupation>, <persname>, <physfacet>, <subject>, <term>, <title>, and <unittype>. These elements could contain URIs encoded for linked data even before EAD3 and this paper follows the development of EAD's support for linked data access points before arriving at EAD3. The second is the incorporation of the <relation> element which had been introduced in EAC-CPF.

Either of these methods may be used at any level of description to describe a relationship. The access point elements may be bundled within a <controlaccess> section or incorporated into other elements which support their use. A <relation> occurs within <relations> directly in the <archdesc> or <c/xx> section of that level of description. Both may fulfill content standard requirements. For example, the required DACS element 2.6, Name of creator(s), could

be encoded within an access point in <origination> or as a <relation>. That same <relation> could also cover DACS element 2.7, Administrative/Biographical History. Because of this functional overlap, the preface to the EAD3 tag library notes that the inclusion of <relation> at all within the standard was controversial within the group.[29] Some considered the ways in which relationships could be expressed through access points and URIs sufficient. Ultimately, <relation> was incorporated into the standard and dubbed "experimental."

For the sake of consistency across finding aids and the importance of predictability when using scripted solutions, those guiding data encoding at a repository should make intentional decisions about the circumstances under which encoders should use access points and/or the relation element. The following sections provide an overview of how one would encode relationships using either method and some guidance on how these relationships would then be extracted as linked data.

Encoding relationships in access points

Through its access point elements, EAD has always provided methods which could be used to encode linked data-ready XML. This was not reflected in the documentation,[30] but institutions have developed local practices entirely consistent with early principles of EAD encoding which allow them to encode and extract linked data.[31] As EAD3 refines and augments these initial methods, rather than inventing its own, an overview of how to encode for linked data in access points must start with EAD 1.0[32] and EAD2002. An institution converting from EAD2002 may have already been encoding in ways which ease its transition to encoding for linked data.

Within the access point elements, EAD 1.0 included (and EAD 2002 retained) the attribute @authfilenumber, which, combined with @source, provides a space to record the authority file record and authority source for a term used.[33] Within the name subset of access point elements,[34] the attribute @role allows one to encode the role or relationship between that

access point and the materials.[35] For example, before the LC Name Authority Files were published as linked data, this is how one would encode author Nnedi Okorafor as the originator of the materials, referencing her LC Name Authority File, in EAD 1.0 and EAD2002:

```
<origination>
<persname role="creator" authfilenumber="n2004028670"
source="lcnaf">Okorafor, Nnedi</persname>
</origination>
```

Although none of this is linked data or even involves URIs, it may be converted to linked data as shown in the next section. It should be noted that the application of an identifier to the entire access point may not be appropriate in all cases. A term used may be more complex than its established form. It may be subdivided by forms, dates, geographical regions, and additional terms. For example, consider the Library of Congress Subject Heading: "Death--Religious aspects--Buddhism, [Christianity, etc.]" which may subdivide by any religion, but is not established as separate terms for these religions. A repository would then have to choose whether to encode the primary term's identifier in @authfilenumber, not use at all, or come up with another method of handling the situation. EAD3 addresses this problem.

Encoding linked data-ready access points in EAD2002

After the Library of Congress began publishing its authorities through its Linked Data Service[36] one could programmatically derive its URIs using the @authfilenumber and source. For example, the URI pattern for lcnaf combines a base URI "http://id.loc.gov/authorities/names/" and the authority file record number "n2004028670" for a complete URI of http://id.loc.gov/authorities/names/n2004028670. If a repository consistently recorded sources and @authfilenumbers, they could extract and construct URIs, then update the original records and/or publish the URIs as appropriately-mapped RDF. For those who do not yet use @authfilenumber, Rubinstein outlines how one might use the Library of Congress's linked data tools to find the appropriate URI for a string.[37]

URIs may be used within @authfilenumber, a practice encouraged by newer tooling. For example, when one adds a subject to ArchivesSpace, the label of the field which generates @authfilenumber "Authority ID" includes hovervtext "the unique identifier for the record within the source from which it was acquired (i.e. an LCSH number). The identifier may be represented by a URI." The ArchivesSpace Public User Interface (PUI) extracts and maps these URIs from into schema.org, expressed as JSON-LD and embedded within the HTML display pages.[38]

The use of the @role provides context for the mapping to RDF. Although this example demonstrates the most straightforward case, a <persname> element not found in <origination> may encode more than the name of the creator of the materials. In order to extract and process data accurately, a script needs more information than that it's handling a <persname>.[39] As with @authfilenumber, one may use URIs within the @role attribute. To reiterate the example above using URIs from the Library of Congress's linked data service, one might express Okorafor's creation of the materials as:

```
<origination>
<persname role="http://id.loc.gov/vocabulary/relators/cre"
authfilenumber="http://id.loc.gov/authorities/names/n2004028670"
source="lcnaf">Okorafor, Nnedi</persname>
</origination>
```

Or one might decide to use the Dublin Core vocabulary and VIAF URI to express the same thing:

```
<origination>
<persname role="http://purl.org/dc/terms/creator"
authfilenumber="http://viaf.org/viaf/78167151" source="viaf">Okorafor,
Nnedi</persname>
</origination>
```

Both statements are correct. Indeed, both are interrelated. VIAF's record states that it is schema:sameAs the LC authority. The LC authority declares that its record is a skos:exactMatch

of the VIAF record.[40] The MARC relator for creator and DCTerms creator are both rdfs:subPropertyOf the original Dublin Core (1.1) creator. Such connections will not always exist, but may influence a data manager's decisions of what predicates to use. After considering the reasons why they are choosing to extract linked data and what they plan to do with those extractions (e.g. enhance search with Schema.org, create Samvera metadata records, etc.) a repository's data manager should set standards for sources of predicates and object URIs.

This in-depth overview of how one may encode for linked data within EAD2002 access points is intended to demonstrate how repositories undertake encoding for linked data before adopting EAD3. EAD3 changes some of the terminology to make the linked data uses of attributes more explicit and fundamentally alters access point encoding in a way that introduces both opportunity and additional complexity for scripted linked data extraction.

Changes to access points in EAD3

In EAD3, access points underwent one minor change and one major change related to linked data encoding. In the minor change, two main attributes used to encode URIs, @authfilenumber and @role were renamed @identifier and @relator respectively.[41] This change brought the terminology in line with the language of linked data and libraries. Additionally, data within them must now comply with the XML "token" restriction (not containing leading/trailing spaces, tabs, line breaks, and other extraneous whitespace). This should not have any effect on a well-encoded authority file number, code, or URI.

EAD3 also introduced the concept of <part> from EAC-CPF. Unlike EAC-CPF, where it may only be used within names, <part> is required within all access point elements in EAD. This allows one to break down names, subject terms, geographic locations, etc. into multiple sections and assign granular data such as @localtype and @identifier at the part level. In EAD3, then, the above example above might look like:

```
<origination>
```

```
<persname relator="http://id.loc.gov/vocabulary/relators/cre"
identifier="http://id.loc.gov/authorities/names/n2004028670"
normal="Okoafor, Nnedi" source="lcnaf"><part
localtype="surname">Okorafor</part><part
localtype="givenname">Nnedi</part></persname>
</origination>
```

Note that, in this case, punctuation has been removed, leaving nothing but raw data of her given name and surname (if she had a date, that would also be encoded separately). The normalized form of her name has been preserved within @normal. Meanwhile, @identifier, used at the level of the access point element, <persname>, ties her name to the authority. This example shows the richest way to encode her name as data, however, encoding her name in full in a single <part>Okorafor, Nnedi</part> is also entirely valid within EAD3.

Besides its utility in breaking down names into data, <part> becomes particularly relevant when dealing with complex subject headings. As earlier in this section, a complex subject heading may not be represented by a single authority file or identifier. If one were dealing with a collection which included materials on Death--Religious aspects--Rider-gods, for example, one would encode the subject as follows:

```
<subject relator="http://purl.org/dc/terms/subject"><part
identifier="http://id.loc.gov/authorities/subjects/sh85036097"
source="lcsh">Death--Religious aspects</part><part
identifier="http://id.loc.gov/authorities/subjects/sh85114014"
source="lcsh">Rider-gods</part></subject>[42]
```

Note that this style of encoding retains punctuation in the established "Death--Religious aspects." Since "Death--Religious aspects--Buddhism, [Christianity, etc.]" may be broken down by any religion, the religion of Rider-gods and its URI are included as a second part. The two subjects extracted together provide more insight into the materials than either no URI or merely a URI representing Death--Religious aspects. A repository's data manager, along with

catalogers and other descriptive workers, should create standards for practice in the use of <parts> and @identifier in any kind of constructed subject.

In some cases, the terms may include elements which have no established term as well as ones which do. Anything which may be subdivided by a date, for example, would only contain @identifier URIs for <part> elements established with URIs. If only a single element is used, the institution might choose to encode @identifiers at the top level, but in cases of multiple subdivisions, one would simply not include @identifier on certain parts. This example would encode a fictional 1985 fire in the Special Collections unit of the Pennsylvania State University Libraries.

```
<subject relator="http://purl.org/dc/terms/subject"><part
identifier="http://id.loc.gov/authorities/names/no2006098295">Pennsylva
nia State University. Special Collections Library</part><part
identifier="http://id.loc.gov/authorities/subjects/sh00005747"
source="lcsh">Fire</part><part localtype="date">1985</part></subject>
```

Without major tools such as ArchivesSpace supporting EAD3 at the time of this writing, aspects of how identifiers in <part> will be supported remains to be seen. As software restrictions often guide practice, the full potential of <part> and a consensus on best practices for its use are unlikely to be seen until major systems of archival description support EAD3.

Programmatically extracting linked data from access points

Everything above has been examples of how to encode linked data-ready XML. In order to extract the actual linked data, one must run the XML through a script (XSLT, Python, Ruby, etc.) or app which extracts the appropriate URIs or older data, parses them, and returns linked data. The example results and assume that the script has gotten the URI of the finding aid from its <recordid>, represented as record:1313. For the following encoding method:

```
<persname role="creator" authfilenumber="n2004028670"
source="lcnaf">Okorafor, Nnedi</persname>
```

The script will need to include a list of URIs associated with values used in @role and a list of URI prefixes associated with values used in @source. The script would then extract "creator" and match it to, for example, the MARC Relators URI for creator, "lcnaf" (finding its associated URI prefix) and "n2004028670" and combine the two to form a single URI. The triple extracted (in Turtle) would then be:

```
record:1313    mrel:cre    lcnaf:n2004028670 .
```

As N-Triples with full URIs, the triple would be:

```
<http://repository.org/1313>
<http://id.loc.gov/vocabulary/relators/cre>
<http://id.loc.gov/authorities/names/n2004028670> .
```

If the script is running against records which already include URIs as @role and @authfilenumber, the same triple could be created by simply extracting the values.

```
<persname role="http://id.loc.gov/vocabulary/relators/cre"
authfilenumber="http://id.loc.gov/authorities/names/n2004028670"
source="lcnaf">Okorafor, Nnedi</persname>
```

For a simple use case in EAD3, one would substitute @relator and @identifier for @role and @authfilenumber. In a more complex case, such as:

```
<subject relator="http://purl.org/dc/terms/subject"><part
identifier="http://id.loc.gov/authorities/subjects/sh85036097"
source="lcsh">Death--Religious aspects</part><part
identifier="http://id.loc.gov/authorities/subjects/sh85114014"
source="lcsh">Rider-gods</part></subject>
```

a script cannot assume where it will find @identifier and should check both the top-level element (if it finds a URI there, it should consider itself done) and check each @part otherwise. @relator should still only be encoded in the access point element. Parsing the example above would lead to:

```
local:1313    dc:subject    lcsh:sh85036097 .
```

```
local:1313    dc:subject    lcsh:sh85114014 .
```

Expressing Relationships in <relation>

<relation>, an entirely new element and way of expressing relationships in EAD3, is derived from the relational elements in EAC-CPF. The introductory material to the EAC-CPF tag library contains an overview of the nature of relations as conceived of by the group. They sought to create records in which the distributed nature of relationships between things could be expressed.[43] Understanding fundamentals of relations in EAC-CPF provides context for understanding the decisions made to create a more flexible <relation> in EAD3 while retaining the @relationtype attribute to create some symmetry. In EAC-CPF, only three types of things may be related to: other corporations, persons, or families (<cpfRelation>); to a function (<functionRelation>),[44] or to an external resource (<resourceRelation>).[45] These three relational elements may then be broken down to a set of much more specific but also entirely defined types through the attribute @[cpf/function/resource]relationtype, each of which has its own restricted list of values.

As mentioned previously, the inclusion of <relation> in EAD3 was debated within the group. One position of those arguing against it was summarized as "that incorporating robust support for Linked Open Data was premature."[46] However, the generic form adopted may allow more robust support of general linked data by not attempting to codify possible uses. In EAD3, one specifies the nature of the more generic element <relation> in its @relationtype attribute, with possible values: cpfrelation, functionrelation, resourcerelation, and otherrelationtype. The fourth value, which should then be declared within @otherrelationtype, allows a repository to define and declare new types of relations to suit its needs. A repository data manager's decision to encode linked data in <relation> should include extensive consideration for how the element as a whole will be used. Do they and other workers believe their collections should connect to types of things in the world beyond cpfs, functions, and resources (and do so through

relations)?[47] Are they able and willing to maintain new relation types? Will <relations>' potential be used in lieu of access points, will it duplicate and flesh them out on occasion, or will it be used only when the statement cannot otherwise be made? The choice to use <relations> opens up a world of possibility but also introduces dependencies in content standard validation, processing for display, and processing for linked data extraction, which should be carefully considered.

Encoding <relation> for linked data

The <relation> element serves as a place for description and display within the finding aid as well as an opportunity to encode for linked data. Rather than go into the use of all of <relation>'s child elements, the following examples will focus on the methods by which one may encode assertions to be extracted as RDF. The simplest way to describe an RDF relationship within <relation> uses only its attributes. <relation> must contain the attribute @relationtype, used as described above, although the data encoded within it is not used for RDF. The following two attributes may then be used to create an assertion which can be extracted as RDF:

- @href - use to encode the object URI.
- @arcrole - use to encode the predicate URI for the triple, such as http://purl.org/dc/terms/creator.

Those two attributes are sufficient to create a simple, functional triple.

The following examples show what it would look like only to use these attributes in <relation> and then what it might look like to use some of its child elements as well. In this example, the hypothetical papers being described belong to Judy Chicago at the Arthur and Elizabeth Schlesinger Library on the History of Women in America at Harvard. A second collection of Judy Chicago's works pertaining to art education exists and is held by the Penn State University Libraries' Special Collections department. The encoder wishes to contribute a value-added DACS 6.3 statement about separated (or related, if these are somehow not

associated by provenance) archival materials. Additionally, the encoder wishes to encode the relationship between Judy Chicago herself and the materials. A simplified version would be encoded as follows:

```
<relations>
      <relation relationtype="resourcerelation"
arcrole="http://www.w3.org/2000/01/rdf-schema#seeAlso"
href="https://libraries.psu.edu/findingaids/9028.htm" />
      <relation relationtype="cpfrelation"
arcrole="http://id.loc.gov/vocabulary/relators/cre"
href="http://id.loc.gov/authorities/names/n78079492" />
</relations>
```

This way of encoding relations is entirely utilitarian and aimed at producing RDF. It does not provide enough information to display in a textual finding aid (although a "see also" or "creator" could be derived from the arcrole).

Or one could encode more in-depth relations intended to be displayed on the page:

```
<relations>
      <relation relationtype="resourcerelation"
arcrole="http://www.w3.org/2000/01/rdf-schema#seeAlso"
href="https://libraries.psu.edu/findingaids/9028.htm" linktitle="Judy
Chicago art education collection at PSU">
            <relationentry>Judy Chicago art education collection at the
Pennsylvania State University Special Collections
Library</relationentry>
            <daterange>
                  <fromdate standarddate="1970">1970</fromdate>
                  <todate standarddate="2011">2011</todate>
            </daterange>
<descriptivenote>
```

```
      <p>This collection of Judy Chicago's work contains materials
related to her art and pedagogy. [... the note might go on to describe
how it differed from the collection being described ...]</p>
</descriptivenote>
      </relation>
      <relations relationtype="cpfrelation"
arcrole="http://id.loc.gov/vocabulary/relators/cre"
href="http://id.loc.gov/authorities/names/n78079492">
            <relationentry>Chicago, Judy, 1939-</relationentry>
            <descriptivenote>
                  <p>From the <ref
href="http://www.judychicago.com/about/biography/">artist's site</ref>:
Judy Chicago is an artist, author, feminist, educator, and intellectual
whose career now spans five decades. [cont …]</p>
            </descriptivenote>
</relation>
</relations>
```

Besides the forms demonstrated in the previous example, <relation> may contain linked data in two other ways. First, before <descriptivenote> it may include a <geogname> element which, as an access point, may contain an @identifier and @relator as described above. <geogname> is the only access point element directly available within <relation> although one may include the rest of the access points within the <descriptivenote>'s <p>s.

Support for complex RDF/XML in <objectxmlwrap>

Finally, <relation> may encode linked data, not using EAD3 elements and attributes, but as RDF/XML encapsulated within its <objectxmlwrap> element.[48] <objectxmlwrap> should contain a single XML root element which then may contain any kind of XML (although it should

not contain EAD3). In order for the EAD3 finding aid to validate, the XML within <objectxmlwrap> must also validate.

Because <objectxmlwrap> may contain any valid XML, it could contain any kind of RDF/XML statement.[49] The assumption of the containing <relation> element is that what it encloses will describe a relationship between materials described in the finding aid it encloses and other things. This would not be the proper place, for example, to include an RDF surrogate for the EAD3 finding aid or a set of RDF statements extracted from access points as described above. Therefore, the object of the statements should be the URI for the collection itself (probably the finding aid's URI). An example from the Judy Chicago section above might be the following (where http://example.org/24.204.10 stands in for the finding aid's URI):

```
<objectxmlwrap>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:mrel="http://id.loc.gov/vocabulary/relators/">
    <rdf:Description rdf:about="http://example.org/24.204.10">
       <mrel:cre dcterms:title="Chicago, Judy, 1939-"
dcterms:description="From the artist's site: Judy Chicago is an artist,
author, feminist, educator, and intellectual whose career now spans
five decades. [cont ...]"
rdf:resource="http://id.loc.gov/authorities/names/n78079492"/>
       <rdfs:seeAlso dcterms:title="Judy Chicago art education
collection at the Pennsylvania State University Special Collections
Library" dcterms:description="This collection of Judy Chicago's work
contains materials related to her art and pedagogy. [... the note might
go on to describe how it differed from the collection being
```

```
described ...]"
rdf:resource="https://libraries.psu.edu/findingaids/9028.htm" />
    </rdf:Description>
</rdf:RDF>
<objectxmlwrap>
```

On <source>'s Potential for RDF Transformation

Another addition from EAC-CPF, the <source>[50] element cannot be left out of an overview of EAD3 and linked data. As an element, it might be lumped in with <relation>, of which it is a near duplicate.[51] However, it occurs within the <control> element, where it is meant to serve as information about the *finding aid* rather than about the *collection* it describes. The element provides the person encoding the finding aid a place in which to add titles of and links to sources which they've consulted in the creation of the finding aid. Statements extracted as linked data should therefore be citations (using such predicates as schema.org's "citation"[52]). Any other relationships between material referenced in <source> and the collection should be described within <archdesc> in any of the manners outlined previously, just as they would have been if it had not been used as reference material.

Programmatically extracting linked data from <relation> and <source>

Actually extracting the linked data from <relationentry> or <sourceentry> requires nothing more than the URI for the collection and a script to find and extract values of @href and @arcrole. Reliance <objectxml> as a site of encoding one's linked data statements is a possibility which simplifies extraction even further. If a repository's finding aid authors were to encode in this way, it should trigger a general re-evaluation of the use of access points and other ways of meeting content standard requirements, as it might otherwise lead to duplication of effort. Although RDF statements must be encoded RDF/XML, they need not necessarily be extracted as RDF/XML. For example, Google recommends the use of JSON-LD when

embedding of schema.org linked data on a webpage.[53] Fortunately, software libraries and tools which transform data between forms of RDF encoding exist for many scripting languages.[54]

Other Forms of Linked Data Extraction

This article sought out to specifically address the parts of EAD3 which made it suitable for encoding linked-data ready description which could be extracted as five-star RDF. However, not all RDF statements must fit this standard. It would be remiss to conclude before briefly addressing how other data encoded within EAD finding aids may be suitable for extraction and presentation as RDF. For example, a collection's title, its description, its dates, and the like may be extracted as text and displayed as objects in RDF statements. One might also extract textual information from the finding aid's <eadheader> (EAD2002) or <control> (EAD3) about the repository itself. For example, the Rockefeller Foundation Archives extracts text about the fundamentals of its collections (such as title, creator, dates created) and text about the Archives itself such as its address and phone number.[55]

As mentioned above, the new ArchivesSpace Public User Interface (PUI) automatically extracts and displays finding aid information in schema.org in order to improve search engine results. In addition to the URIs mentioned above, the PUI also extracts text even if URIs exist (as additional context) or in elements which would not have a URI encoded. Those at institutions using ArchivesSpace and primarily concerned with Schema.org may consider this a compelling reason to focus on encoding for linked data in access points rather than waiting for its adoption of EAD3 and <relation>.

Conclusion

With new and traditional authorities being offered in linked data format, the work being done within the archival community map to linked data standards, and the developments within EAD3 which improve its support for linked data, the current technical landscape is an appealing prospect for archivists. Whether currently encoding within EAD3 or planning to move to it at

some point in the future, they may still take advantage of the support within the standard and begin the progression. Those who had already been regularly encoding authority file numbers may now see a path toward refining their own data for the semantic web. Even if the repository lacks support to adopt linked data any time soon, encoding good data in access points as described above will allow future archival workers to transform and use that data. If all an archival worker can control is the quality of their data, they can still encode linked data-ready description.

Notes

[1] Society of American Archivists Technical Subcommittee for Encoded Archival Description, *Encoded Archival Description Tag Library - Version EAD3*, (Chicago: Society of American Archivists, 2015), available and cited at http://www.loc.gov/ead/EAD3taglib/index.html, accessed September 5, 2017.

[2] Wim van Dongen, report to the Technical Subcommittee for Encoded Archival Standards. (Presented at Society of American Archivists Annual Meeting, Portland, Oregon, July 26, 2017).

[3] Society of American Archivists Technical Subcommittee for Encoded Archival Description, Preface.

[4] The schema.org ontology was developed by a partnership including major search engines for better interchange between a variety of data standards and a format which they could recognize and index. http://schema.org/docs/about.html

[5] Proposal submitted by Richard Wallis on behalf of the Schema Architypes W3C Community Group at https://github.com/schemaorg/schemaorg/issues/1758, accessed September 29, 2017.

[6] This paper references ArchivesSpace exclusively when providing examples of software used for URI encoding in archival description. It is one of the few undergoing ongoing development and, in the author's location of North America, the only one of those to allow for the encoding of URIs and publication of those URIs as linked data.

[7] Mark A. Matienzo, Elizabeth Russey Roke, and Scott Carlson, "Creating a Linked Data-Friendly Metadata Application Profile for Archival Description" (Poster presentation at DCMI International Conference on Dublin Core and Metadata Applications, Washington, DC, October 26-29, 2017). arXiv:1710.09688 / http://dcpapers.dublincore.org/pubs/article/view/3860

[8] Daniel Pitti, Bill Stockting, Florence Clavaud, "Records in Contexts (RiC): a standard for archival description developed by the ICA Experts Group on Archival Description." https://www.ica.org/en/records-in-contexts-ric-a-standard-for-archival-description-presentation-congress-2016, accessed September 23, 2017

[9] Society of American Archivists, Encoded Archival Description Working Group, *Encoded Archival Description Application Guidelines for Version 1.0 , (Encoded Archival Description (EAD), Document Type Definition (DTD), Version 1.0, Technical Document No. 3)*, (1999). Available and cited at http://www.loc.gov/ead/tglib1998/tlprinc.html, accessed September 9, 2017.

[10] *Ibid*., http://loc.gov/ead/tglib/elements/runner.html, accessed September 9, 2017.

[11] *Ibid*., http://loc.gov/ead/tglib/elements/frontmatter.html, accessed September 9, 2017.

[12] *Ibid*., http://loc.gov/ead/tglib/elements/persname.html, accessed September 9, 2017. The entry's description of @role also reads as though the author had MARC relators and/or Dublin Core on their mind.

[13] *Ibid*., http://loc.gov/ead/tglib/att_gen.html, accessed September 9, 2017.

[14] Or RDF, a series of W3C recommendations available at https://www.w3.org/standards/techs/rdf.

[15] Ed Jones, "Introduction," in *Linked Data for Cultural Heritage*, ed. Ed Jones and Michele Seikel (Chicago: ALA Editions, 2016), x.

[16] National Information Standards Organization. *Issues in Vocabulary Management*, (Baltimore: NISO, 2017), http://www.niso.org/apps/group_public/download.php/18410/NISO_TR-06-2017_Issues_in_Vocabulary_Management.pdf, accessed September 27, 2017.
[17] Jones, xi.
[18] A string literal (referred to in casual documentation as either a “string” or a “literal”) is a set of characters, including letters, numbers, and special characters. In a program, these are enclosed in quotation marks and treated as a functional unit, much as one would treat multiple words, dates, and symbols in a term from a controlled vocabulary as a unit. When an access point is text, rather than URI, it is a string literal from a controlled vocabulary.
[19] James Kim and Michael Hausenblas, “5-star Open Data,” *5-Star Open Data*, http://5stardata.info/en/, accessed September 12, 2017.
[20] The examples in the *Records in Context - Conceptual Model* draft v0.1 demonstrate a conception of graphs as simply an alternative to encode text, not as an opportunity to create linked data. None include URIs as the triple’s object. https://www.ica.org/sites/default/files/RiC-CM-0.1.pdf, accessed, September 21, 2017.
[21] The technical specification for Turtle can be found at https://www.w3.org/TR/turtle/, accessed September 18, 2017.
[22] Although the Samvera URI Working Group’s Predicate Decision tree was designed for use within a particular software ecosystem, its list of common ontologies may also be useful for archivists
https://wiki.duraspace.org/display/samvera/URI+Management+Working+Group?preview=/87460991/87462917/PredicateDecisionTree.pdf, accessed September 22, 2017
[23] One aspect of the linked open web is that anyone may say anything about anything. Multiple viewpoints about the same Thing may be represented. Or malicious, misinformed, or otherwise inaccurate assertions may cause improper inferences to be drawn. This should not deter cultural heritage organizations from engaging and attempting to contribute the best data possible.
[24] Karen F. Gracy, “Archival description and linked data: a preliminary study of opportunities and implementation challenges,” *Archival Science* 15 (2015): 239-294.
[25] Hillel Arnold, “Implementing Schema.org at the Rockefeller Archive Center,” *Bits and Bytes* (blog), October 17, 2013, http://blog.rockarch.org/?p=826, accessed September 8, 2017.
[26] Winona Salesky and Hillel Arnold, *XTF-RAX*, https://github.com/RockefellerArchiveCenter/XTF-RAC, accessed September 8, 2017.
[27] Aaron Rubinstein, “Sharing Archival Metadata,” in *Putting Descriptive Standards to Work* (Chicago: Society of American Archivists, 2017).
[28] Ed Jones and Michele Seikel, *Linked Data for Cultural Heritage* (Chicago: ALA Editions, 2016).
[29] Society of American Archivists Technical Subcommittee for Encoded Archival Description, Preface.
[30] ...or necessarily in the intention of its creators.
[31] For example, Princeton’s award-winning finding aid site allows one to view its finding aids as XML or RDF by appending .xml and .rdf respectively. One may see examples similar to those in this section in its EAD2002 finding aids and how they are then extracted as RDF.

[32] Access points in EAD 1.0 differ, in some cases, from those in EAD2002. However, since the relevant attributes @authfilenumber, @source, and @role were all present in EAD 1.0, their inclusion from the beginning should be acknowledged.
[33] Society of American Archivists, Encoded Archival Description Working Group, http://www.loc.gov/ead/tglib1998/tlatt1.html
[34] <corpname>, <famnname>, <geogname>, <name>, and <persname>
[35] Society of American Archivists, Encoded Archival Description Working Group, http://www.loc.gov/ead/tglib1998/tlatt1.html
[36] Library of Congress, *LC Linked Data Service: Authorities and* Vocabularies, http://id.loc.gov, accessed September 5, 2017.
[37] Rubinstein, 339-340.
[38] The Public User Interface release 2.1.0 includes these changes. https://github.com/archivesspace/archivesspace/releases/tag/v2.1.0, accessed September 24, 2017.
[39] In fact, other than matching the element to extract the URIs, the script does not need to know that what it's handling is a personal name.
[40] Note that the two choose different ontologies to express a very similar type of relationship between their records and thus the statements are equivalent only as much as the similar types of relationship are nearly equivalent.
[41] This is distinct from @arcrole, which will be noted and used in the section on <relation>.
[42] Although these examples use @relator within subject, a repository's data manager may decide non-name access points have default URI mappings (such as always mapping <subject> to http://purl.org/dc/terms/subject) and make them part of scripted extraction.
[43] Technical Subcommittee for Encoded Archival Context of the Society of American Archivists, *Encoded Archival Context—Corporate Bodies, Persons, and Families (EAC-CPF Tag Library)*, 2014, Relations, http://eac.staatsbibliothek-berlin.de/fileadmin/user_upload/schema/cpfTagLibrary.html#d0e650, accessed September 20, 2017.
[44] Generally to be considered as defined in the International Standard for Describing Functions (ISDF), which may then be constrained to "functions of corporate bodies associated with the creation and maintenance of archives." from https://www.ica.org/en/isdf-international-standard-describing-functions, accessed September 8, 2017.
[45] Any kind of resource, from an archival collection to a catalog record for their work to a digitized edition of that work and more. Of the attributes within relations, @resourcerelationtype is the only to include the value "other."
[46] Society of American Archivists Technical Subcommittee for Encoded Archival Description, Preface.
[47] Note that the prefix of the relation speaks to its type of Thing, rather than the type of relation between those two things.
[48] The <objectxmlwrap> element was introduced from EAC-CPF and is not supported for those using the XML DTD instead of the XML Schema or RNG.
[49] Although N-Triples and other methods of encoding RDF may resemble XML, only RDF/XML will validate within the <objectxmlwrapper> element.

[50] Society of American Archivists Technical Subcommittee for Encoded Archival Description, <source>, http://www.loc.gov/ead/EAD3taglib/index.html#elem-source, accessed September 17, 2017.
[51] Although <source> does not include child elements to encode dates or geographic information separately from the <descriptivenote>.
[52] https://schema.org/citation
[53] "Introduction to Structured Data," *Google Developers*, https://developers.google.com/search/docs/guides/intro-structured-data, accessed September 20, 2017.
[54] Because of the variety of languages in which they exist and the speed with which such tools may be written and become obsolete, this paper does not make recommendations on transformational tooling.
[55] Arnold, "Implementing Schema.org at the Rockefeller Archive Center."