Towards multiscale archival digital data

Laurent Romary, Charles Riondet

Towards multiscale archival digital data[1]

Laurent Romary (Inria & DARIAH), Charles Riondet (Inria)

In this paper, we would like to present some ideas on the use of the archival standards in various contexts that exemplify the complexity of such standards and provide users with innovative ways to handle EAD content. Our main idea is that researchers, Cultural heritage institutions, archival portals and standards maintenance bodies could greatly benefit from a multiscale modelling of archival data, but also from multiscale representations and documentations. A first step is on the way to being cleared in the domain of the management of heterogeneous archival sources in one single environment, namely a federated portal, like in EHRI. We built a methodology based on a specification and customisation method inspired from the long lasting experience of the Text Encoding Initiative (TEI) community. In the TEI framework, one has the possibility of defining project-specific subsets or extensions of the TEI guidelines while maintaining both the technical (XML schemas) and editorial (documentation) specification within a single framework. Using the same framework for EAD data allows us to express precise content-oriented rules combined with some interesting possibilities of integrating the human readable documentation in the validation process.

EAD, What's wrong with it

The development of EAD was initiated in 1993 at the Library of Berkeley, with the idea of building a non proprietary format for finding aids, reflecting the hierarchical structuration of archival fonds. If preliminary attempts were expressed in SGML, the first version of EAD used XML, and was released in 1998. A second version was released soon after in 2002, EAD2002, which is still the most widely used version. It is maintained by the Library of Congress and the Society of American Archivists. In 2010, a global revision process was initiated, in order to make EAD more connected to Linked Data technologies, and to reach a better integration with the others XML archival formats: EAC-CPF and EAG and in 2015, EAD3 was officially released.

However, in the world of cultural heritage institutions and research, archival description is often considered as a pending issue, a hindrance to data exchange and accuracy. Since its creation, EAD faces criticism, as many observers are pointing to its permissiveness as a problem. Yet in 2001, Shaw asks for a "more prescriptive descriptive standard" [Shaw, 2001]. Still today, and even if EAD3[2] is globally seen as a step in the right direction, EAD is generally seen as a poorly structured and interoperable standard, not very suitable for data exchange, and is paradoxically considered by some information specialists, not a “standard for archival description” [Bunn, 2013]. We will not go any further in this controversy, but point the fact that the archival community, though aware of these weaknesses, still broadly works with EAD and is still willing to improve the quality of digital archival descriptions.

[1] This work is developed in the context of the H2020 projects EHRI and PARTHENOS

[2] https://www.loc.gov/ead/EAD3taglib/index.html

There is room to improve EAD in two main aspects: 1) handle its flexibility and 2) preserve all the complexity of the content when exchanging archival description. Of course, the new *Records in Context* content mode [EGAD 2106] proposes a nice way to handle these issues, with an ontology meant to bring together all the pieces of archival information (authorities, institutions, functions and records), natively compliant with semantic web technologies. But, before this solution is adopted and implemented, EAD still is and will be the archival community standard. The framework we propose will allow for better exchange and dialog between archival data and together with others resources available online.

The EHRI use case

The EHRI environment is a perfect use case to apply our method, because of the heterogeneity of the corpus, characterized by a great diversity of languages, description levels, and archival practices, and the goal to ingest all these archival descriptions in one single environment. These various sources need therefore to be compared, checked in quality, and processed before being integrated in the repository. To do so, the pivot format is naturally EAD (version 2002), used for automatic ingestion in EHRI database and also for exports. Like for all the archival portals, the two crucial questions are how to deal with so many differents ways of encoding EAD, and how to guarantee that the descriptions are compliant with EHRI requirements. To handle this situation, we propose a method to create customizations for EAD in order to refine archival descriptions both in the structure and in the content, and of course respect entirely the EAD syntax.

This method is developed in the context of the umbrella project Parthenos[3] which aims, among other things, at disseminating information and resources about methodological and technical standards in the humanities. One of the main objectives of Parthenos is to create a Standardization survival Kit (SSK) [Romary et al., 2016], whose main features are to:

- Propose generic research scenarios to scholars where the use of standards play a key role
- Communicate around community initiatives
- Support standardization activities in domains where it is needed.

Within Parthenos, one of the scenarios we will provide in the SSK is precisely a scenario guiding scholars and cultural heritage information specialists in the creation of project-specific EAD schemas.

The TEI-ODD specification framework

In this project, we are inspired by another very strong community standard : the Text Encoding initiative. This format facilitate the representation of any textual resource in XML. It was built for digital editions of historical texts, but can be used in many other situations. For instance, what we are interested in is a subset of the TEI meant to create XML formats specification (the TEI itself is described with this subset of TEI). This is called “One document does it all” and it allows us to model specific subsets, extensions or profiles of the described format. ODD can be used to refine the behaviour of elements and attributes, for

[3] http://parthenos-project.eu

any XML format, contains all the human readable documentation and can be processed to generate various resources: a validation schema (in many formats) and some documentation (in many formats).

ODD is based on the principles of literate programming, which means that this language combines formal (specifications) and informal declarations (descriptive prose and examples) [Knuth 1983]. It combines in the same environment the technical specifications and the user guidelines for the key components of the TEI Abstract Model, primarily elements and attributes, but also modules, classes and macros [Burnard & Rahtz 2004]. For example, to write the specification of an element, the tag used is `<tei:elementSpec>`. It contains elements for documentation, like the `<tei:gloss>` (a phrase or word used to provide a gloss or definition) or `<tei:desc>` (a brief description of the object documented by its parent element, typically a documentation element or an entity). The `<tei:classes>` element is used here to link elements with their attributes, and the `<tei:content>` contains the relaxNG specification, *i.e.* what elements can be children of the described element.

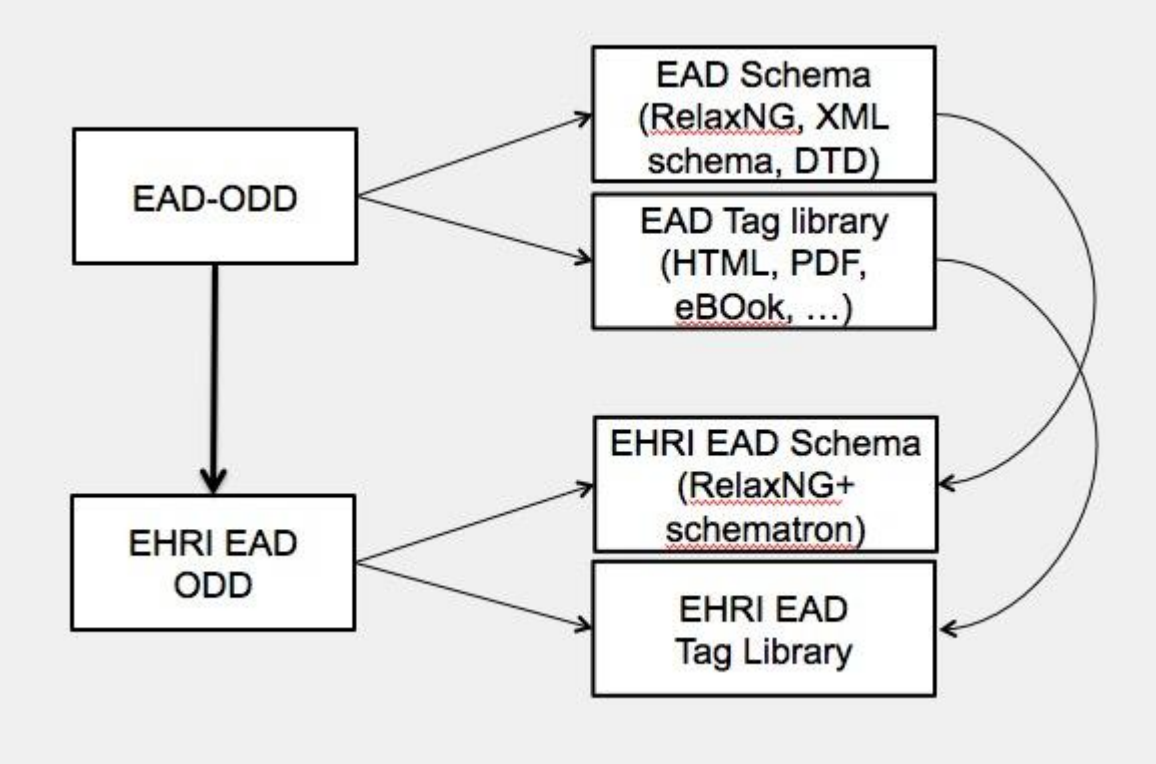
```
<elementSpec ident="c01" module="EAD">
    <gloss>Component (First Level)</gloss>
    <desc>A wrapper element that designates the top or first-level subordinate
        part of the materials being described. Components may be either
        unnumbered <gi>c</gi> or numbered <gi>c01</gi>, <gi>c02</gi>, etc. The
        numbered components <gi>c01</gi> to <gi>c12</gi> assist a finding aid
        encoder in nesting up to twelve component levels accurately.</desc>
    <classes>
        <memberOf key="att.EADGlobal"/>
        <memberOf key="att.desc.c"/>
    </classes>
    <content>
        <rng:optional>
            <rng:ref name="head"/>
        </rng:optional>
        <rng:ref name="did"/>
        <rng:zeroOrMore>
            <rng:ref name="model.desc.full"/>
```

Fig. 1: *Example of the ODD specification for an EAD element*

The official EAD schema and the official EAD tag library were encoded in an ODD document (Agreement of the Library of Congress and the Society of American Archivists), in the context of the Parthenos project[4]. This EAD ODD is a starting point for EHRI, used to create an EHRI-specific EAD profile with very precise content oriented rules based on EHRI requirements and on the CHI data models and some qualitative documentation to be served to the user of conversion and validations services provided by the EHRI project.

4

http://github.com/ParthenosWP4/standardsLibrary/blob/master/archivalDescription/EAD/odd/EADSpec.xml



EHRI specific rules

EHRI has its own ODD, project specific, that inherits everything from the generic EAD ODD, except the elements and attributes that have a different behaviour in EHRI. The philosophy is to keep the EAD schema as it is, i.e. not modify directly the RelaxNG specifications. Instead, we use another validation language: ISO Schematron[5]. EHRI already used schematron rules to control the input descriptions. We completed them, respecting the same organisation. Schematron validation serves diagnostics to the content providers, by emphasizing:

- technical errors and proposes a solution, as EHRI conceive it
- EHRI descriptions guidelines requirements
- EHRI descriptions guidelines proposals, or “nice to have” points

Some rules reflects the requirements of EHRI database content model. For instance, it asks that the `<date>` elements contains a `@normal` attribute whose content respect the ISO8601 standard on representation of dates and time. This constraint is expressed in the ODD file with embedded schematron in the following way:

```
<elementSpec ident="date" module="EAD" mode="change">
    <constraintSpec ident="dateNormal" scheme="isoschematron" type="EHRI" mode="add">
       <desc>All the <gi>date</gi> elements MUST have a <att>normal</att> attribute whose
pattern respects the ISO8601 standard and take the following form: YYYY-MM-DD</desc>
    <constraint>
               <sch:rule context="date">
                      <sch:assert role="MUST"
test="matches(@normal,'^(([0-9]|[1-9][0-9]|[1-9][0-9]{2}|[1-9][0-9]{3}))-(0[1-9]|1[012])-
(0[1-9]|[12][0-9]|3[01])$')">@normal attribute MUST respect ISO8601 pattern = YYYY-MM-
DD</sch:assert>
```

[5] http://schematron.com/

```
            </sch:rule>
    </constraint>
</constraintSpec>
</elementSpec>
```

This second rule is also a requirement, but for different reasons. For the sake of comprehension of the archival description, EHRI requires that a `<scopecontent>` element should be present somewhere. the choice is let to the provider to write on general paragraph and put it at the highest level (`<archdesc>`) or add a more precise `<scopecontent>` for each subcomponents, from `<c01>` to `<c06>`. Here, the rule is called at the `<archdesc>` level, because it is more likely that the CHI provides a global `<scopecontent>` if it didn't exist before.

```
<elementSpec ident="archdesc" mode="change">
<!-- … -->
    <constraintSpec ident="scopecontentInArchdescOrC" scheme="isoschematron" type="EHRI">
            <desc>A <gi>scopecontent</gi> element SHOULD be present in the description
       at least in <gi>archdesc</gi>, if not in the <gi>c</gi> elements.</desc>
            <constraint>
                <sch:rule context="archdesc" role="SHOULD">
                        <sch:assert test="scopecontent or dsc/c01/descendant-or-
                self::scopecontent">a "scopecontent" element SHOULD be present at
                least in "archdesc" if not in the "c" elements</sch:assert>
                </sch:rule>
             </constraint>
    </constraintSpec>
</elementSpec>
```

The last rule showed is the lowest level of constraint. It presents some possibilities to make the description more complete. In particular, these rules focuses on the content related elements of `<archdesc>`. Therefore, these messages are not considered as real errors, but as pieces of advice that the providers can follow or not.

```
<elementSpec ident="archdesc" mode="change">
<!-- … -->
    <constraintSpec ident="bibliographyPossible" scheme="isoschematron" type="EHRI">
            <desc>The <gi>archdesc</gi> element COULD contain a <gi>bibliography</gi>
             element.</desc>
                    <constraint>
                   <sch:rule context="archdesc">
                          <sch:assert  role="COULD"  test="bibliography">archdesc  COULD
have a bibliography</sch:assert></sch:rule>
                    </constraint>
          </constraintSpec>
```

The rules added to the EAD schema in EHRI specify all the different parts of the archival description : the administrative metadata (the `<eadheader>`, in particular the history of the modification of the EAD), the description itself (`<archdesc>`, `<c>` and `<did>`), and the content elements (the access points, i.e. the named entities, persons, places, organisations, …, but also the dates). Another type of specific rules is related to all the standardized codes

used to identify some pieces of information, like the languages used (ISO639), the archives (ISO15511), etc.

Overview of EHRI schematron rules:

1. Administrative metadata (<eadheader>)

EHRI Rules	**Role**
<eadheader> must contain a <profiledesc> element	MUST
the <eadid> element must contain text. Most of the time, it is automatically generated by the archival tool.	MUST
<eadheader> must contain information on the language used in the EAD document with a <langusage> element containing a <language> element	MUST
<eadheader> should contain a <creation> element	SHOULD
<eadheader> should specify a <publisher>	SHOULD
The <titlestmt> element should contain a <author> element	SHOULD
A date of creation for the finding aid is welcome. The relevant element is <date>, child of <creation>	COULD
<descrules> has a default value added automatically by EHRI. Therefore, the content of <descrules> will be overwritten	
In <revisiondesc>, each <change> element should contain a <date> element and a <item> element. [This rule has been taken from Library of Harvard Archivesspace Checker[6]]	SHOULD
The <date> element for each <change> in <revisiondesc> should not be empty	SHOULD

2. Archival description (<archdesc>, <did>, <c>)

EHRI Rules	**Role**
<archdesc> must have a level attribute.	MUST
The value of the <archdesc> @level attribute should be limited to four values: - fonds	SHOULD

[6] https://github.com/harvard-library/archivesspace-checker/blob/master/schematron/archivesspace_checker_sch.xml

- recordGrp - collection - otherlevel	
<dsc> must have a @type attribute	MUST
if <dsc>'s @type attribute has "othertype" for value, <dsc> must have a not empty @othertype attribute	MUST
The sub components elements (<c01> to <c06>) must have a @level attribute.	MUST
If the @level attribute of <c01>, <c02>, … has the value 'otherlevel', an attribute @otherlevel MUST be added	MUST
<did> elements must contain: - a <unitid> - at least on non-empty <unittitle>	MUST
Each unit of description should have an identifier in the element <unitid>.	SHOULD
In a given EAD document, all the <unitid> elements must be unique	MUST
In the <did> element, <physdesc> should come with a non-empty <extent> element	SHOULD
<archdesc> should contain a non-empty <origination> element.	SHOULD
<archdesc> should contain a non-empty <processinfo> element.	SHOULD
The <processinfo> element should contain a <date> element as descendant.	SHOULD
A <scopecontent> element should be present in the description at least in <archdesc>, if not in the sub components elements.	SHOULD
The sub components elements should be numbered components between <c01> and <c06>	SHOULD
The <archdesc> element could contain these content related elements: - <langmaterial> - <custodhist> - <otherfindaid> - <originalsloc> - <altformavail> - <bibliography> - <odd> - <note> - <controlaccess>	COULD
<langmaterial> could contain a <language> element.	COULD

If the element <altformavail> is not empty, you could try to identify if the originals are present in the EHRI portal and make a link between the two descriptions.	COULD
If the element <originalsloc> is not empty, you could try to identify if copies are present in the EHRI portal and make a link between the two descriptions.	COULD

3. Entities

a. Access points

EHRI Rules	**Role**
In <controlaccess>, EHRI welcomes any access points types : <subject>, <geogname>, <persname>, <orgname>.	COULD
Access points could be chosen in authority lists. The list is declared with a @source attribute. The related identifier of this authority should be declared in an @authfilenumber attribute. Note that EHRI provides URLs for vocabularies and authorities.	COULD
In the access points, person names should be structured like this : Family name, given name	SHOULD

b. <unitdate> and <date>

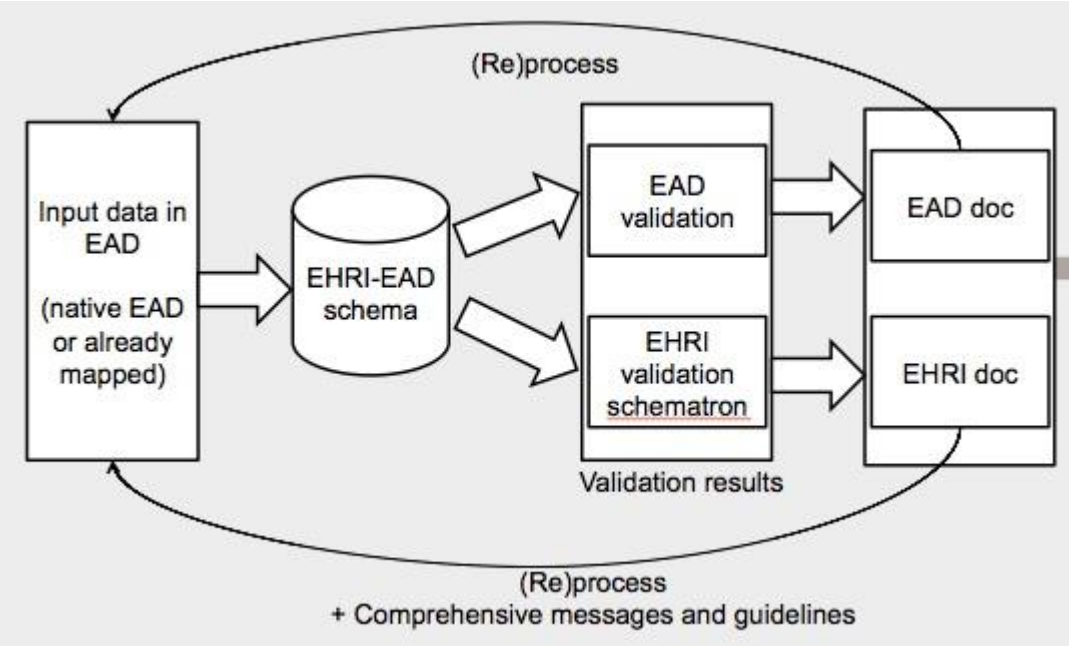
EHRI Rules	**Role**
<unitdate> should have a non-empty @normal attribute	SHOULD
The @normal attribute of <unitdate> must respect the ISO8601 pattern: YYYY-MM-DD	MUST
<unitdates> could have a @label attribute or an @encodinganalog attribute, describing the type of date	COULD
All the <date> elements must have a @normal attribute whose pattern respects the ISO8601 standard and take the following form: YYYY-MM-DD	MUST

4. Languages and other coded values

EHRI Rules	**Role**
<language> must have a @langcode attribute, taken from the list given by the ISO639 standard.	MUST
<language> must have a @scriptcode attribute, taken from the list given by the	SHOULD

ISO15924 standard.	
If the language of the description is not English, a parallel form of the title in English should be added. For instance, using another `<unittitle>` element with a `@type` attribute	SHOULD
`<eadid>` should contain a `@mainagencycode` attribute, which provides (if applicable) the ISO15511 code for the institution that maintains the finding aid.	SHOULD
If the `@repositoryencoding` is set to iso15511, the format of the value of the `@repositorycode` attribute is constrained according to the International Standard Identifier for Libraries and Related Organizations (ISIL: ISO 15511): a prefix, a dash, and an identifier.	SHOULD

In the EHRI mapping and validation workflow, the EHRI-EAD schema is used to test the archival descriptions before they are ingested in the portal. The result is of this validation is a list of messages (presented above) linked to precise fragments of the tested description. Therefore, the archive that ingests its descriptions in EHRI portal is informed of the changes it has to make to be sure its data could be integrated in the portal harmlessly. In the future, it is also planned that some uncritical modifications could be automatically made inside the validation framework (base on the Schematron Quickfix extension[7]).



[7] http://schematron-quickfix.github.io/sqf/publishing-snapshots/April2015Draft/spec/SQFSpec.html

Conclusion

Offering a standard-based method to gain interoperability between heterogeneous data allows users, above all researchers, to access high quality standardized data. On the other hand, a small CHI sharing easily its data via the EHRI portal gains visibility, by showing easily underexposed data, and creates data enrichments opportunities.

This method may be of a wider interest within similar environments (i.e. archives portals). As it is one of the components of the Parthenos *Standardization Survival Kit* - a solution that offers researchers needing standardized methods and resources complete frameworks to carry out their project, in Arts and Humanities and Heritage science, it can be used freely by any interested project. Parthenos is also willing to give support and maintain the EAD ODD for a substantial period.
More, this solution can be seen as a possible bridge between EAD2002 and EAD3, and more broadly could be considered as a tool for the future maintenance of the EAD standard, in order to, like for the TEI, oriente this maintenance towards a (wise) ever ongoing revision methodology.
It could also be an opportunity to bring together EAD and TEI and propose on the fly generation of skeletal TEI documents based on EAD descriptions.

Bibliography

Bunn, Jennifer. 2013. “Developing Descriptive Standards: A Renewed Call to Action.” *Archives and Records* 34 (2): 235–47. doi:10.1080/23257962.2013.830066.

Burnard Lou, Rahtz Sebastian, 2004, "RelaxNG with Son of ODD", *Proceedings of Extreme Markup Languages 2004*. : http://conferences.idealliance.org/extreme/html/2004/Burnard01/EML2004Burnard01.html

Experts group on archival description (ICA). 2016. “Records in Contexts, a Conceptual Model for Archival Description. Consultation Draft v0.1.” Conseil international des Archives. http://www.ica.org/sites/default/files/RiC-CM-0.1.pdf.

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. doi:10.1093/comjnl/27.2.97.

Shaw, Elizabeth J. 2001. “Rethinking Balancing Flexibility and Interoperability.” *New Review of Information Networking* 7 (1): 117–31. doi:10.1080/13614570109516972.

Romary, Laurent, Degl’innocenti Emiliano, Illmayer Klaus, Joffres Adeline et al.. Standardization survival kit (Draft). Deliverable 4.1, written by members of PARTHENOS WP4. 2016. <hal-01513531>