

Major revision of EAD

[1. Introduction](#)

[2. A general note to start with](#)

[3. Alignment with EAC-CPF](#)

[3.1. Introduction of camelCase spelling](#)

[3.2. Aligning the definition of shared elements](#)

[3.2.1. <control>](#)

[3.2.2. Date elements <date>, <dateRange>, and <dateSet>](#)

[3.2.3. Elements to encode places and place names](#)

[3.2.4. <relations> and shared entity elements](#)

[3.2.5. In-line tagging for shared elements](#)

[3.3. Aligning the definition of shared attributes](#)

[3.3.1. Attributes for internal and external referencing](#)

[3.3.2. Other changes regarding attributes](#)

[4. Describing various instances of a finding aid](#)

[5. Relations and relationships](#)

[6. Formatting and in-line tagging](#)

[7. Additional changes for existing elements](#)

1. Introduction

The revision of the [Encoded Archival Description \(EAD\)](#) was approved by the Society of American Archivists' (SAA) Council according to the SAA's standards revision cycle at the end of 2020. The work of revising EAD is being carried out by a team of the Technical Subcommittee on Encoded Archival Standards (TS-EAS), a group supported by SAA and overseen by their Standards Committee. The EAD team has been working on the revision since early 2021 to the current date (end of February 2024), which has led to the creation of an initial draft for EAD 4.0.

It is now nearly nine years since the release of the previous major revision, EAD3, hence EAD 4.0 aims to **bring EAD up to speed with current developments** in the world of archival description as well as the wider context of publishing cultural heritage collections online. Specifically:

- This revision looks at **aligning EAD with** its sibling standard, the [Encoded Archival Context - Corporate Bodies, Persons, and Families \(EAC-CPF\)](#), to enable a smoother side-by-side use of both, as well as with other standards often used in relation to EAD. This not only relates to the encoding options, but also to the **terminology used in the names of elements and attributes**, which have been adapted to use a more generally applicable and accessible language. The alignment specifically with EAC-CPF also lays a lot of groundwork for other changes applied in EAD 4.0, which is the reason why these changes make a significant part of what is described below.
- The revision includes **EAD's relation to** [Records in Contexts \(RiC\)](#), version 1.0, which was released in December 2023. RiC replaces the existing ICA standards such as the General International Standard Archival Description (ISAD(G)), for which

EAD currently provides an XML-based implementation approach. Re-evaluating EAD's relation to the ICA standard hence becomes pivotal. The Encoded Archival Standards (EAS) may, as a whole, in their future versions (among which EAD 4.0) be considered an XML-based representation of the RiC Conceptual Model alongside the RiC Ontology.

- Furthermore, this revision looks at how EAD can better support all the developments in terms of **Linked Open (Usable) Data**, which makes the Internet tick nowadays. While EAD3 already does this to some extent, possibilities for improvements have been brought to the table during conversations with the community throughout the past years.

This current document groups the changes applied in the draft for EAD 4.0 into four main areas and is meant to accompany the [XSD](#) and [RNG](#) schema and other supporting documentation during the call for comments. While the descriptions of the changes below will not include all the details, links have been included to the respective issues on GitHub that provide those details as further background, if required. It should be noted that the current draft version of EAD 4.0 is the result of many in-depth conversations and explorations that the EAD team of the Technical Subcommittee on EAS (TS-EAS) has had during the past three years, even if not all of them will be mentioned specifically. This includes:

- EAD's relation to other standards such as PREMIS, MARC, BIBFRAME, etc.;
- A series of use cases for EAD that resulted from some of the earlier community engagement activities;
- Tag usage statistics as generated e.g. in the context of the [NAFAN](#) project (over 145,000 EAD XML files including nearly 27 million component descriptions from twelve state and regional level aggregators across the United States) and by the [Archives Portal Europe](#) (nearly 680,000 EAD XML files including over 300 million component descriptions from all over Europe).

2. A general note to start with

As announced when EAD3 was published, all elements and attributes that were deprecated in EAD3, have now been removed from EAD 4.0 ([#52](#)). Furthermore, EAD 4.0 will replace some more elements and attributes by alternative ways of encoding.

3. Alignment with EAC-CPF

During the major revision of EAC-CPF, which led to the publication of [EAC-CPF 2.0](#) in August 2022, the Technical Subcommittee on Encoded Archival Standards (TS-EAS) started working on an alignment between the standards it maintains. This meant that all elements and attributes named and used in the same way in both EAC-CPF and in EAD also needed to be defined in the same way in both standards. A big part of the revision process has hence been a review on how elements and attributes are currently defined in EAC-CPF and EAD. Especially for those elements and attributes with a shared definition, this also included a confirmation that these are indeed used in the same way. The resulting alignment of how these elements and attributes are defined, has led to the following changes for EAD 4.0.

3.1. Introduction of camelCase spelling

A shared definition not only refers to the sub-elements and attributes that can be used with a specific element or the data type for a specific attribute, but also to how this element or attribute is named and the way how that name is spelled. EAD 4.0 will hence follow EAC-CPF 2.0 in applying camelCase spelling to any element and attribute names consisting of a combination of terms, e.g. <biogHist> from “biography or history” (#49). This also adheres to best practices in XML definitions and makes the XML tagging easier to read for the human eye, especially considering the broad international community of EAD users, many of whom do not speak English as their first language. Better readability also supports better recognisability of an element or attribute’s name’s origin and makes EAD easier to learn.

With this change agreed, there also came the opportunity to review the names of elements and attributes more generally for better alignment with the terminology used in related standards such as ISAD(G), RiC, or Describing Archives: A Content Standard (DACS), but also in collection management systems that support EAD as an exchange format (#66). In this context, for example, the elements <accessrestrict> and <userrestrict> have been renamed to <accessConditions> and <useConditions>. Similarly, this brought the chance to review abbreviated names of elements and attributes and to consider more concrete and more easily understandable names (#53). It should be noted that not all abbreviated names of elements and attributes have been changed, as TS-EAS felt that some of these have been established and well-understood for two decades now. However, changes such as renaming <acqinfo> to <sourceOfAcquisition> or <odd> to <otherDescriptiveInfo> fall into this category.

3.2. Aligning the definition of shared elements

3.2.1. <control>

In terms of shared elements, the <control> element might be the most prominent one. EAD 4.0 now has <control> and all its sub-elements defined in the same way as EAC-CPF 2.0 (#32) and even includes some additional changes, which will later on be aligned in a next revision of EAC-CPF. The following is only a selection of changes, but covers the most important ones:

- The sequence of sub-elements of <control> has been adapted, so that all mandatory elements (<recordId>, <maintenanceAgency>, and <maintenanceHistory>) are now listed first. This should be of help especially in any hand-coding context, where the current set-up in EAD3 with e.g. <maintenancehistory> only appearing at the end can be misleading.
- <recordId> cannot be left empty anymore. This is especially important when EAD is used as an exchange format, where the identifier of the EAD XML file itself often plays a pivotal role in ingest and update scenarios.
- <otherRecordId> now includes the option to link to the context from where this other record identifier has been taken, making use of the three new optional vocabulary attributes @valueURI, @vocabularySource, and @vocabularySourceURI.
- The use of the sub-elements of <maintenanceAgency> has been loosened up by requiring either <agencyName> or <agencyCode> to be present. For anyone working with persistent identifiers for institutions, this provides a more consistent way to refer to the agent responsible for the creation and maintenance of the EAD XML file.
- The <source> element now uses the general element <reference> (renamed from <ref>) to refer or link to the source in question. Furthermore, a new sub-element <citedRange>

has been added to enable a more precise indication of the source material e.g. by specifying a certain page or page range.

- Some elements (<maintenanceStatus>, <publicationStatus>, and <eventType>), that were used with fixed values in EAD3, have been turned into optional attributes and can now be used with either a suggested “EAS list” of values or one’s own list, which is recommended to be declared using the element <conventionDeclaration>.
- In <conventionDeclaration>, <localTypeDeclaration>, and <rightsDeclaration> the element <citation>, which is only used on this specific context in EAD3, has been merged with the general element <reference>, which allows the user to do exactly the same thing. Furthermore, the element <abbreviation>, which also is only used in this specific context in EAD3, has been renamed to <shortCode>.
- The element <languageDeclaration> has been streamlined by moving the attributes @languageCode and @scriptCode directly into <languageDeclaration> and removing <language> and <script> (#38). Being a sub-element of <control>, the use of standardised values in the attributes will allow for more automation respectively will help in avoiding typos or variations of abbreviations, which could result in contradicting language statements, when hand-coding.
- The element <filedesc>, which represented the biggest difference between <control> in EAD3 and EAC-CPF 2.0, has been moved out of <control> and is now an optional and repeatable sibling element called <findAidDesc>, which sits between <control> and <archDesc> (#27). More details on this change is in section [4. Describing various instances of a finding aid](#) below.
- All predefined value lists for attributes have been removed from the EAD 4.0 schema (#1), allowing the respective attributes to be used with any literal values. In the next steps of this major revision, a separate vocabulary of “EAS lists” will be made available including the TS-EAS recommended values along with some explanation of what each value means and how the values within each list relate to each other. This vocabulary, if used, can then be referenced via a series of new optional “...Encoding” attributes within <control>. Alternatively, it will be possible to use one’s own values and vocabularies by indicating the use of an “other...Encoding” scheme with these attributes. This category e.g. includes the three new attributes @maintenanceStatus, @publicationStatus, and @maintenanceEventType mentioned above, but also attributes such as @audience, @addressLineType, @level and @unitDateType. Existing ways of capturing alternative values have been folded into this new approach, resulting e.g. in the removal of attributes such as @otherlevel and in the removal of @localType when available next to one of the attributes in this category, as e.g. in <addressLine>.

3.2.2. Date elements <date>, <dateRange>, and <dateSet>

EAD 4.0 adopts the encoding of dates as defined in EAC-CPF 2.0 by focusing on the triad of <date>, <dateRange>, and <dateSet> (#4). This also means the removal of the distinction between <date> and <dateSingle>, which is made in EAD3 based on the context in which the elements are used, while they functionally do the same thing. The single date elements, i.e. <date>, <fromDate>, and <toDate>, now all have the same set of optional attributes. This includes a new @status attribute to e.g. indicate when a date is unknown, the attributes @calendar, @era, and @certainty to include a literal expression of the context and certainty - or uncertainty - of a date, and only the attribute @standardDate (instead of alternatively allowing the attribute @normal). @standardDate can be used with the newest version of the ISO standard 8601 (integrating the [Extended Date/Time Format \(EDTF\)](#)), which enables

standardised uncertain and approximate date qualifiers, or with any other standard for encoding dates as chosen by the maintenance agency of the EAD XML file.

Similarly, the elements <dateRange> and <dateSet> use the same set of attributes, adding the new reference attributes (see section [3.3. New optional attributes for internal and external referencing](#) below) as well as allowing for @localType and the new parallel attribute @localTypeDeclarationReference. Furthermore, <dateRange> now requires either <fromDate> or <toDate> to be present, while both could be left empty, for example, using the @status attribute to indicate that the one or the other or both are “unknown”. If both sub-elements are used, <fromDate> is required to appear before <toDate>.

3.2.3. Elements to encode places and place names

EAD 4.0 introduces the element <place> as a broader concept of encoding places, their names, roles, types, geographic location and further details ([#42](#)). <place> will require at least one of the following sub-elements to be present, though leaves it to the users of EAD to decide which of these they make use of: <placeName>, <placeRole> (e.g. place of creation or place of access), <placeType> (e.g. implementing the [feature codes used by GeoNames](#), such as “populated place” or “first-order administrative division”), <geographicCoordinates> (thereby elevated to being a sibling rather than a sub-element of the element used to encode the name of a place), <address> (for any physical address details), and the new element <contact> (for any digital contact details).

<placeName>, <placeRole>, and <placeType> all allow for a literal expression of the name, role or type of place, but also include the three new vocabulary attributes @valueURI, @vocabularySource, and @vocabularySourceURI (see more in sections [3.3. New optional attributes for internal and external referencing](#) and [5. Relations and relationships](#)) enabling the inclusion of terms from externally maintained vocabularies. <geographicCoordinates> remains more or less unchanged, apart from the camelCase spelling and some new optional attributes. The new <contact> element has the same set-up as the existing <address> element with a <contactLine> sub-element and the only difference, that <contactLine> also allows for the linking attributes @href, @linkRole, and @linkTitle.

<place> will be available to its full extent within <findAidDesc> (see section [4. Describing various instances of a finding aid](#)) and <relations> (see section [5. Relations and relationships](#)), while <placeName> will be used in other contexts of EAD 4.0 instead of the element <geogname> of EAD3. Contrary to <geogname>, <placeName> will hold any literal text directly, without the sub-element <part>. Furthermore, the plural element <places> has been adopted from EAC-CPF and is available next to the other entity elements as described in section [5. Relations and relationships](#).

3.2.4. <relations> and shared entity elements

The element <relations> and its direct sub-element <relation> will be aligned with how they are defined in EAC-CPF 2.0 ([#33](#)). For <relations>, this mainly means the addition of the optional sub-element <descriptiveNote>. For <relation>, in order to make sure the related entity is at least named, the optional sub-element <relationentry> is replaced by the required sub-element <targetEntity>. Furthermore, the required attribute @relationtype is transformed into an optional element <relationType>. Thereby the use of predefined values is removed, instead allowing for any literal text within the element itself plus enabling the three vocabulary attributes @valueURI, @vocabularySource, and @vocabularySourceURI to

include relationship terms from externally maintained vocabularies. Next to categorising the relationship as a whole, the type of the related entity and the role that this related entity has towards the materials described can be encoded with the new optional elements `<targetType>` and `<targetRole>`.

Similarly, the shared elements `<function>` ([#61](#)) and `<legalStatus>` ([#5](#)) will be defined in the same way as they are in EAC-CPF 2.0. For `<function>`, this means a change from the sub-element `<part>` to `<term>` as well as the addition of a series of optional sub-elements used with other entity elements as well (see `<place>` as described above and more in the section [5. Relations and relationships](#)). `<legalStatus>`, which is a general descriptive element in EAD3, will follow more or less the same content model by also requiring a `<term>` sub-element and allowing for the encoding of a temporal and geographical dimension related to the legal status of the materials described. While falling under the entity class of rules, `<legalStatus>` is understood to be more of a characteristic of the materials described, which also fits with the definition of the element in previous versions of EAD. As such, `<legalStatus>` has been moved into `<identificationData>` (renamed from `<did>`).

For more on relations and relationships see [section 5](#) of this document.

3.2.5. In-line tagging for shared elements

In-line tagging, or mixed content, has always been one of the bigger differences between EAD and EAC-CPF. Aligning both standards means finding a common approach to this, which still enables EAD to support in-line tagging as much as possible while not overburdening EAC-CPF.

One change in this category is the integration of the element `<emph>` for marking words or phrases that are to be emphasised or formatted differently with the element `` ([#71](#)), which comes with a similar but at the same time slightly broader definition and also uses a more open approach to alternative formatting options with the attribute `@style` (instead of EAD's `@render` attribute).

Another change is the introduction of a more generic mixed content model, where `` is accompanied by the element `<reference>` (combining EAD's current `<ref>` and `<ptr>` elements) and the new element `<referringString>` (allowing users to tag named entities within textual descriptions). This mixed content model will be applied to shared elements such as `<abstract>`, `<p>`, and `<reference>` ([#47](#)) as well as a series of EAD-specific elements (see more on this in section [6. Formatting and in-line tagging](#)).

3.3. Aligning the definition of shared attributes

3.3.1. Attributes for internal and external referencing

EAD 4.0 will pick up on several attributes that EAC-CPF 2.0 either has newly introduced or extended in their scope. Most of these relate to possibilities for referencing other elements within the same EAD XML file or for referencing external vocabularies providing additional context.

For generic internal referencing from one element of an EAD XML file to another, the attribute `@target` has been added to all elements in EAD 4.0, which was bound to the availability of either `<ref>` or `<ptr>` in EAD3. Furthermore, the attributes data type has been

changed to IDREFS now allowing for referencing more than one element of the same EAD XML file at once ([#31](#)).

Similar to `@target`, the new attributes `@conventionDeclarationReference`, `@maintenanceEventReference`, `@sourceReference` (see [#11](#) for all three), and `@localTypeDeclarationReference` ([#43](#)) can be used to specifically point to:

- `<conventionDeclaration>` or `<localTypeDeclaration>` elements which encode information about rules or standards or local value lists that have been used in the creation of the EAD XML file;
- `<maintenanceEvent>` elements which encode information of what changes were made when and by whom;
- `<source>` elements which encode any additional source for certain information used in the description of the archival materials.

The attributes `@conventionDeclarationReference`, `@maintenanceEventReference`, `@sourceReference` will be available with all elements in the descriptive parts of EAD 4.0, while `@localTypeDeclarationReference` is used in all elements that allow for the addition of a `@localType`. With its broad availability, `@conventionDeclarationReference` specifically can also be used instead of the attribute `@transliteration` to point to any transliteration rules applied to the content of any descriptive element and specified in `<conventionDeclaration>` ([#34](#)) and instead of the attribute `@rules` available with the `<controlaccess>` sub-elements and some others in EAD3 for mentioning any descriptive rules applied to the content of an element ([#13](#)).

Next to these attributes for internal referencing, EAD 4.0 will adopt the three vocabulary attributes `@valueURI`, `@vocabularySource`, and `@vocabularySourceURI` as introduced in EAC-CPF 2.0. It should be noted that the term “vocabulary” is to be understood as broadly as possible in this context. It is not only bound to the most internationally used Linked Open Data vocabularies such as the Library of Congress Subject Headings, the Virtual International Authority File, GeoNames, or Wikidata, but also includes any defined and maintained list of codes and/or terms, whether these are used only within one institution or on a bigger scale.

In terms of EAD, this means renaming the attribute `@identifier` to `@valueURI` and adapting its data type to the slightly stricter “anyURI”. It also means changing the attribute name `@source` to `@vocabularySource` (which also follows the TS-EAS design principle of not having an element and an attribute of the same name) and adding the new attribute `@vocabularySourceURI` for cases where the vocabulary used is publicly available online ([#13](#), [#51](#), and [#72](#)). The change in data type for `@valueURI`, allows the attributes `@xpointer` and `@entityref` to be removed from EAD ([#45](#)).

3.3.2. Other changes regarding attributes

In addition to the changes mentioned in the previous section, EAD 4.0 will follow EAC-CPF 2.0 in renaming `@lang` and `@script` to `@languageOfElement` and `@scriptOfElement` ([#30](#)) as well as `@langcode` and `@scriptcode` to `@languageCode` and `@scriptCode` ([#38](#)). Furthermore, EAD 4.0 will introduce the option to include attributes from any other namespace ([#28](#)) to accommodate any categorisation, typing, or formatting specifications that users might want to encode, but that are not part of EAD - or will not be part of EAD 4.0 anymore. For instance, this could facilitate the use of attributes from the XLink namespace,

specifically `@xlink:actuate`, `@xlink:arcrole`, and `@xlink:show`, as the respective attributes `@actuate`, `@arcrole`, and `@show` from EAD3 will be removed ([#41](#)). This is in line with other changes aiming at focusing EAD on encoding archival descriptions and not on encoding how these are displayed (see section [6. Formatting and in-line tagging](#)).

4. Describing various instances of a finding aid

As mentioned above (see section [3.2.1. <control>](#)), EAD 4.0 moves `<filedesc>` out of `<control>`, renames it to `<findAidDesc>` to avoid the “file” (which can have multiple meanings and is hence not considered clear enough), and makes `<findAidDesc>` an optional and repeatable sibling element of `<control>` and `<archDesc>` ([#27](#)). The intent is to keep `<control>` focused on administrative information necessary for the management and maintenance of the EAD XML file, while `<findAidDesc>` captures information about any instantiation of the finding aid, whether this might be a printed version, an HTML page, a PDF document, or publication information about the EAD XML file itself. As such, `<findAidDesc>` is more comprehensive than `<filedesc>` and incorporates functionality that was separated out to the element `<representation>` and to the attribute `@instanceurl` with `<recordid>` in EAD3.

In addition, `<findAidDesc>` makes use of elements also applied in other parts of the EAD XML file, such as `<agent>`, `<citedRange>`, `<date>`, `<formattingExtension>`, `<place>`, and `<title>`, rather than having a series of elements only used in this specific context, which were not functionally similar if not the same.

5. Relations and relationships

Besides aligning the element `<relations>` and its sub-element `<relation>` with their definitions in EAC-CPF 2.0, EAD 4.0 also integrates these elements more directly into existing elements that represent related entities and their relationship to the materials being described. Instead of being a separate addition to the element `<archDesc>` and numbered and unnumbered `<c>` elements as in EAD3, `<relations>` becomes more of an optional extension to existing elements. The intent is to create the opportunity for a relations model that can be extended step-by-step in the way an institution with archival holdings feels comfortable with and capable of. This approach allows progression:

- From providing literal values in the descriptive elements
- Via adding URIs to external resources providing more information about the related entities in the attributes `@valueURI` and `@vocabularySourceURI`
- To describing the relationship between these related entities and the archival materials in `<relations>`, `<relation>` and its sub-elements.

The approach suggested in the draft for EAD 4.0 is the result of discussions within the EAD team of TS-EAS about relations and relationships in EAD that spread over more than two years and had several iterations until arriving at the current stage. EAD 4.0 will include three elements that represent related entities which are considered heightened importance when describing archival materials:

- `<agents>`, which incorporates the agent acting - or having acted - as the creator of the materials, the agent acting as the current maintenance agency for the materials, and agents having played any other type of role in the creation, maintenance, or use of the materials ([#58](#));

- <functions>, which these actors fulfil - or have fulfilled - in creating, maintaining, or using the materials ([#61](#));
- <formsAvailable>, which incorporates any number of digital as well as analogue forms in which the materials are available, including the availability of copies either at the institution describing the materials or at any other institution ([#65](#)).

To indicate the importance of these entities' relations with the materials being described in EAD, the concept of plural and singular elements as initially introduced in EAC-CPF will be applied ([#68](#)) next to existing concepts in EAD. <agents>, <functions>, and <formsAvailable> all require at least one singular element of <agent>, <function>, or <formAvailable>, and allow for the optional element <descriptiveNote>.

<function> ([#61](#)), which is moved out of the context of <controlaccess>, will be adapted to the element's definition in EAC-CPF 2.0 and require the sub-element <term> (instead of <part>). Furthermore, <function> allows for encoding a date and place dimension and re-uses the element <targetRole>, <targetType>, and <relationType> from the element <relation>.

<agent> ([#58](#)) will be used with a required <agentName> sub-element to pick up on predecessor elements such as <corpname>, <famname>, <name>, and <persname>. It will also allow for an <agentRole> and an <agentType> to be defined and will additionally include elements for encoding a date and place dimension as well as re-using the element <relationType> from the element <relation>.

<formAvailable> ([#65](#)), which integrates elements of EAD3 that follow a slightly different content model, will allow for a textual description using the elements <abstract>, <p>, or <formattingExtension> at first. Furthermore, it will include <relations> as a whole to enable a more specific encoding of the relationship between the materials being described and its variety of instantiations.

Similarly to <formAvailable>, <relations> will also be included in those elements that represent archival as well as bibliographic resources that are related to the materials being described such as <otherFindAid>, <publicationNote> (renamed from <bibliography>), <relatedMaterial>, and <separatedMaterial> ([#69](#)).

All elements mentioned so far will additionally allow for the use of the three attributes @valueURI, @vocabularySource, and @vocabularySourceURI for referencing any external descriptions of the related entities. These three attributes will furthermore be available in a variety of other elements, which could possibly be represented with a URI, to enhance EAD's adaptiveness to Linked Open Data and Semantic Web use cases. Elements allowing for the three vocabulary attributes include:

- The shared elements <agencyCode>, <agencyName>, <conventionDeclaration>, <legalStatus>, <localTypeDeclaration>, <maintenanceAgency>, <otherAgencyCode>, <otherRecordId>, <placeRole>, <relationType>, <rightsDeclaration>, <source>, <targetEntity>, and <targetRole> ([#5](#), [#32](#), [#33](#), [#42](#));
- EAD-specific elements that allow for @identifier, @source, and other attributes in EAD3 such as <physFacet>, <subject>, <title>, <unitId>, <unitTitle> and <unitType> ([#13](#), [#51](#));
- Other EAD-specific elements such as <container>, <archDesc> and the numbered and unnumbered <c> elements, <materialSpec> ([#72](#)).

The broad availability of the vocabulary attributes supported the decision to remove the attributes `@encodinganalog` and `@relatedencoding` (#37). Instead of, for example, giving a MARC code in `@encodinganalog` and specifying MARC as the related standard in `@relatedencoding`, this can now be encoded using `@valueURI` and `@vocabularySource` respectively. Contrary to EAD3, where `@relatedEncoding` was only available as an “apply to all” option in the elements `<ead>`, `<control>`, and `<archdesc>`, `@vocabularySource` is now available right next to `@valueURI`, allowing for a more detailed and more diverse usage of related standards.

As a last “relational” element, the new element `<subjectHeadings>` will be introduced, integrating the existing elements `<controlaccess>` and `<index>`. `<subjectHeadings>` will allow for `<abstract>`, a choice between `<p>` and `<formattingExtension>` and one or more `<subject>` elements (#67). A `@localType` attribute will be available to distinguish between different types - or usage scenarios - for `<subjectHeadings>`. The sub-element `<subject>` will follow the same content model as `<function>`, i.e. requiring one or more `<term>` sub-elements and enabling the encoding of a date and place dimension, a `<targetRole>`, a `<targetType>`, and a `<relationType>`.

6. Formatting and in-line tagging

As mentioned previously, EAD 4.0 will introduce a reduced mixed content model for in-line tagging to enable a more aligned use of EAD with its sibling standard EAC-CPF, specifically in terms of shared elements, but also regarding other elements. A new element `<referringString>` will be added to enable the tagging of named entities or concepts within textual descriptions (#63). Together with `` (integrating `<emph>`, #35) and `<reference>` (renamed from `<ref>` and integrating the elements `<ptr>`, `<archref>`, and `<bibref>`, #40, #46), `<referringString>` will be used as the only mixed content model in EAD 4.0 rather than a variety of mixed content models (#47, #70).

Furthermore, EAD 4.0 will focus on encoding archival descriptions rather than encoding how these descriptions are to be displayed. For this, EAD 4.0 will add the new element `<formattingExtension>` as a choice for more elaborate formatting (in the form of lists or tables or by allowing for several, hierarchically related headers, etc.) next to simple `<p>` elements (#62). `<formattingExtension>` will support any elements of the XHTML namespace to do so. While this means that EAD 4.0 will not include any more extensive formatting elements anymore in its own schema, it also means that EAD 4.0 will enable even broader formatting possibilities than EAD3 for those who want to make use of `<formattingExtension>`. In addition, this approach is expected to help in making sure that these formatting elements always represent the latest status of HTML encoding (on which EAD originally based most of the formatting elements currently available in EAD3) and that there is an easier integration for descriptive and formatted texts with any kind of software for display as well as for management of archival data.

Both of these changes extend on what is available in EAC-CPF 2.0 at the moment and will be aligned again in a future revision of EAC-CPF.

7. Additional changes for existing elements

In addition to the changes described as part of the four main areas above, there are just a few further noteworthy changes to existing elements:

- <dateRange> and the three vocabulary attributes will be made available with <accessConditions> and <useConditions> ([#60](#));
- <abstract> will be moved out of <did> and into longer descriptive elements like <biogHist> or <scopeContent> ([#57](#)), which to a certain extent is also a change in terms of alignment with EAC-CPF;
- Nesting will be removed from <biogHist>, <scopeContent>, etc. ([#59](#)).