TS-EAS shared schema considerations

Technical Decisions

When modularizing the disparte schemas under TS-EAS's maintenance, we will need to make the following decisions:

Which namespace to use

We have minted an EAD3 namespace, an EAD2002 namespace, and an EAC-CPF namespace. We do not have an EAS namespace. We also do not have an official EAF namespace yet.

For the first act of harmonization, we might want to utilize one primary namespace per document that we expect to be delivered as a stand-alone XML file -- i.e. not change anything. Therefore, EAC-CPF would still be in its own namespace, separate from the EAD namespaces. That said, we might want to update the current EAC namespace to use an "http" format rather than an "urn" format. If we change the EAC namespace with the major revision, then we should consider using an EAS namespace at that time with the expectation that we could (if desired) use a single namespace for whatever schema that TS-EAS provides (e.g. EAD3, EAC-CPF, EAF, EAG, etc. could all be in the EAS namespace).

In short, we can continue exactly as is and use whatever namespace we want, but there should be a considerable simplification by selecting one XML namespace for our entire domain.

Which naming conventions to use

(camelCase vs. lowercase vs. PascalCase vs snake_case for elements, whatever else for attributes, etc.)

If we decide to keep separate naming conventions for EAC and EAD3 (e.g. eac:biogHist and ead:bioghist), then all elements will be defined using the camel-case (PascalCase or snake_case) naming convention. The next time that EAD is revised, then the schema-publication process would lower-case those names for the EAD3 schema deliverables. Or, we could switch to camel-casing in EAD with the release of EAD3 2.0 (a 2.0 release of EAD3 would be a major release which need not be backwards compatible).

There are more options than that, of course -- and I might suggest going with PascalCase for elements, and snake_case for attributes -- but it would be simplest given the community's history to either: a) use camel-casing for every TS-EAS schema (elements and attributes); or b) use camel-casing in EAC/F/G and lowercasing exclusively in EAD; or c) lower-case everything in EAC/F/G.

That said, if we continue to have separate naming conventions, then I do not think that we could choose to deploy a single EAS namespace. I would think that having something like <eas:bioghist> and <eas:biogHist> would make our domain look like a very, very confused domain. A single namespace in that case would certainly lead to implementation errors, but I also think that such an inherent, yet misleading, similarity between EAD and EAC speaks to the need to have a single naming convention, so I hope that we would not choose option "b" as defined above.

How far to go with the harmonization process

Using a modular approach along with schema annotations, we have the option to define shared elements in different ways within our different schema deliverables. For just one example, even if we define the control element once, we could allow EAD's profile of the control element to include a filedesc element, whereas the EAC version would continue to exclude filedesc. Alternatively, if we wanted a fully-harmonized control section, then we could either opt to promote <filedesc> to become a sibling element of <control> in EAD3 2.0 or consider adding <filedesc> as sub-element of <control> in the next version of EAC-CPF.

Since modularizing the schemas will make the schemas extensible, it will be up to us to decide how similar (or not) we would like to keep each of the modules. Therefore, whether we have any interest whatsoever to allow an attribute like @audience in EAC or not, or whether we want EAD3 to benefit from improvements that are made during EAC's major revision or not, either decision is possible to accommodate. That said, the more similar that EAD and EAC become, the easier that those standards will be to adopt (by users, tools, etc.), maintain, and to teach.

Which schema serialization(s) do we publish

There are six different schema serializations published for EAD3, along with a Schematron file. That is not only difficult to maintain, but it is also difficult for users (and other potential implementers) to navigate and understand. EAC has published three different schema serializations, but there are significant differences between the Relax NG variants and the XSD variant (e.g. the issue that @xml:id is defined one way in the Relax NG schemas and another way in the XSD schema). However, it seems that the majority of other communities provide one and only one schema serialization format (e.g. METS provides http://www.loc.gov/standards/mets/mets.xsd and that's it).

Ideally, we would provide the same schema serializations for each schema, and only one serialization for each schema. We could decide, for instance, to provide one Relax NG schema with embedded Schematron rules for each standard. Whatever we provide beyond one schema includes costs, so we should make sure that the community requires those additional deliverables before continuing to pay those costs.

What Else? :)

TS-EAS's Schemas Subteam should also establish a set of first principles to guide schema design, especially as we undertake a major revision of EAC and potentially introduce a schema for EAF. For example, we could have guidelines in place for when to model an entity as an attribute versus an element, never to reuse the same entity name as both an attribute name and an element name (regardless of which schema), etc. And the decisions that we make during this process will provide the foundation for such a set of principles.

We should also consider the impacts to tag library maintenance and the question of how a shared schema would be managed on GitHub.