**Assignment 3 Write-Up**

Working with PREMIS was interesting, and a very detailed, complex experience. The schema had a number of challenges, but they mostly seem to result from a drive to enable as much preservation data to be stored as possible, as well as flexible, reactive schema modification for a given repository. However, whenever the goal is a generalizable, adaptive product, there will be a substantial learning curve and lack of clarity without basing new records off of existing records, themselves records for the very same type of item, at the same level of specificity, and in the same repository.

PREMIS has a huge number of fields, and a very open, un-rigid structure and set of fields that allows for immense (relative to what this class has covered thus far) level of tailoring to a given repository. I could see PREMIS working well for preservation data for literally any collection of items. This is atypical for most of the schemas we’ve examined thus far, though perhaps not unexpected for PREMIS, which seems to act as a container schema, allowing a breadth of information to be stored, but without necessary, *explicit* depth. But the interesting, and for-this-class-thus-far novel attribute comes in the ability to fold in other metadata schemas that enable increased, specific metadata fields that allow for deep description for items of various types. Using PBCore to cover the A/V specifics in a way that is standard for both one’s own repository and for others, and without creating your own local controlled vocabulary is an immense benefit. For me, this is the metadata schema equivalent of wishing for more wishes, and really allows endless tailoring of PREMIS to any preservation repository that exists now and into the future. PREMIS seems to allow the inclusion of endless metadata fields in the <significantProperties> field, but this field has no specific controlled vocabulary associated with it, and isn’t a standalone set of xml that can be easily mined and extracted. PBCore provides both, and is ubiquitous enough that transitioning a repository’s collection into PREMIS should be quite a bit easier than other schemas, as PBCore (or any other metadata-specific schema) already existing for items can be extracted and simply pasted into (or better yet, programmatically added) to the PREMIS file.

Additionally, the open vocabulary allows for a repository to use their own custom vocab, for when tracking an object or a value is most important in a more narrow sense—such as within parts of a single repository facility—or even in a broader sense, between locations of the same repository system, or for items that might travel to other repositories or museums to be exhibited, restored or put in storage. For this reason, it seems a great choice for most museums to employ as their standard schema for item metadata, depending on the level of expertise present in the staff. Other schemas are easier to digest, but if you have the expertise on staff, PREMIS would allow greater depth to description and broader coverage for a diverse collection of items. I think this could resemble utilizing PREMIS for general preservation information about objects, information about agents and rights, and then links between objects in the repository. Under each object’s description, a domain- or object-specific schema could be employed, such as PBCore or MIX for film or static images, to better preserve specific characteristics of a given item.

Ironically though, many of these positives of PREMIS can also be negatives, depending on the situation. For this assignment, working on a standalone record without a collection of other items to compare it to, the learning curve for this schema is pretty steep. Even if one could consult other records, the customizability of the schema is such that it can still be hard to learn what’s best for a given object for which you are using PREMIS to record metadata. Outside of an assignment setting, this would be eased partially, of course, but I imagine it would still take a lot of learning, brainstorming and then design to get a record where it should be to best record all of the preservation and provenance metadata necessary to adequately preserve the item. This is new for our purposes, as most of the schemas we’ve looked at have such rigid structures and definitions, meaning the largest conceptual or philosophical choice comes in the very choosing of the schema, as compared to the tailoring process of said schema.

My last major criticism of PREMIS might be a small quibble overall, but this schema is intensely more difficult than the others we’ve looked at when it comes to human reading and parsing. Besides the presence of potentially numerous unrelated objects in a single PREMIS file, the file itself for only one item is very long. Yes, there are longer schemas (MARC records for a full volume with page level information can be pretty tedious), but I found the PREMIS element names relatively unintuitive and the records get very long, in comparison to many other schemas.

Overall, this assignment was pretty challenging, and seemed to be so mostly due to PREMIS having unique characteristics and capabilities. However, I do feel like this would be a fantastic schema for preservation metadata for a repository, and the learning curve would be worth it in the end, given the level of detail and customization provided to the user and the repository. I do feel that PREMIS affords more than MIX or PBCore on their own, especially since either schema could be folded into PREMIS records for items, where appropriate. METS seems to offer a lot of the same benefits as PREMIS, though, and seems customizable as well, if perhaps not quite as much.