Findings of location, container and barcode management in ArchivesSpace

Summary recommendation - Rachel

Results of our testing and data modelling work indicate that ArchivesSpace + the Yale container plugin partially meets the basic location, container and barcode management needs of NYU Libraries. Of 37 requirements, 21 were fully supported by the application, 12 were not supported by the application, and 4 were either partially supported or produced inconclusive testing results.

Additional plugin development is proposed to meet the following requirements:

No	Requirement	Priority (Must, Should, Would)
10	When deleting container(s), ability for a pop-up to ask if associated extents and instances should also be deleted.	W (Or eliminate from list?)
14	The ability to assign or update dimension and capacity information about the locations themselves in order to create location profiles that operate similarly to container profiles. This would require additional fields to be added to the location record.	M
15	Ability to assign or update particular “owners” or functional activities related to a specific location (eg - shelves devoted to access copies, holding area for offsite pick-up, or a particular repository)	S
19	Ability to generate a report that shows available empty space so that archivists can identify where to shelve new material.	M
20	Ability to generate and download a shelf list report by a specific location or location range	M
22	Ability to generate and download a shelf list report for a specified repository (if using one AS instance)	M
31	Ability for archivists to batch assign multiple locations to multiple boxes in a single action or interface [option #3]	S

The following requirements were identified as being unsupported or partially supported, but through discussions on potential policy decisions and alternative options, we will not request further development at this time:

No	Requirement	Suggested Alternative

4	Possible request for change to data model to address insufficiency of instance type and possibly link instances with extent statements [discuss options]	Address with consistent descriptive policies...
7, 24	Ability to add column(s) to table in container interface showing linked records so that collection-level information like call number and/or collection title can be included (eg - for use when viewing or generating a list of items within a specific shared box)	This can be configured in-house.
17	Ability to assign temporary locations without erasing other location information	This has been identified as a bug with an outstanding JIRA ticket (AR-1197).
18	Identify a container as “missing”	Create location called “Missing” that can be applied upon identifying a container as missing, although policy decisions would still have to be made regarding whether and how to document the intended location.
21	Ability to generate a shelf list report for a specific collection	Users can search for this information through the interface and print the results, if necessary. Another option is to query the database.
23	Ability to generate a report by container profile	Users can search for this information through the interface and print the results, if necessary. Another option is to query the database.
25	Ability to generate custom reports ad hoc for other purposes that combine multiple aspects of location and container information	We can use SQL queries for these cases, and while nice, this does not constitute core functionality that we absolutely need.
26	Ability to generate a spreadsheet of locations for movers or for deliveries	This does not constitute core functionality that we absolutely need.
27	Ability to generate box labels with minimal formatting	We will be able to either develop a process that addresses this or find a different way to create box labels.
28	Ability to export MARC xml file that can be loaded into Aleph with minimal human manipulation (that includes location, barcode, and container information)	Our MARC record needs regarding processes and workflows extend beyond location, container, and barcode management.
32	Possible request for ability to import data from a csv or xml file [discuss options, community development]	Between an existing JIRA story and other development in the community, we will be able to develop a process for these.
37	Sensible link between container information for accessions and associated resources.	Can be addressed through policy and workflow decisions

Requirements identification

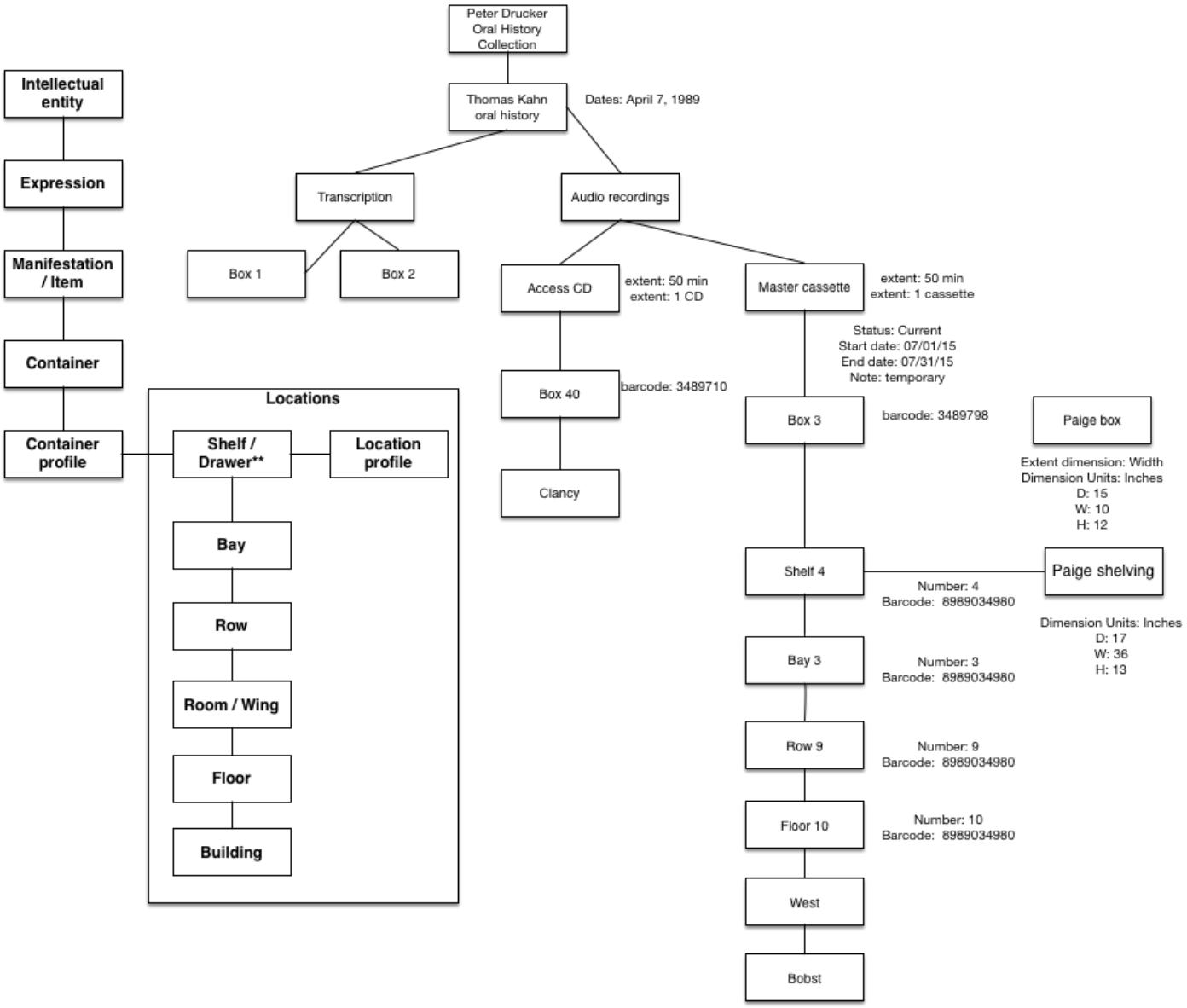
To generate requirements and/or user stories we used the following methods:

- **Discount / rapid requirements approach**. As a place to begin, we brainstormed requirements for a location and barcode management solution. The initial requirements generated were useful as they set a rough scope for further investigation and capture some blue-sky functionality. A mapping of these initial requirements against Archivists Toolkit, ArchivesSpace, and a plugin being developed by Hudson Molonglo for Yale was attempted but was found to be too high-level to provide sufficient confidence that we had captured all use cases.
- **Contextual interviews with Special Collections**. Detailed discussions about current location, barcode and container management processes were held with archivists at Fales, TamWag and UA to confirm current processes. From these and the initially brainstormed requirements, we generated a set of Requirements and a few examples / specific use cases. These requirements defined what data needs to be captured, reporting, batch or bulk operations, import functions, and barcode functions.

Data model comparison - Sally

The following draft model represents our current ideas about what would be the **ideal** data model for location, container and barcode management:

Strawman of ideal data model for location, barcode and container management information*



*not canonical UML - just a quick sketch; also have omitted some attributes for simplicity
**could consider a 'top location', though bay, row etc are actually larger than shelves

Key features of the ideal data model that appear to be changes to the current data model[1] include:

- The ability to create location profiles to associate with specific locations - these profiles would include dimension information to allow for automated calculation of the amount of empty shelving available. An alternative, simplified solution to support our use cases could be to include a field for each location that indicates how many boxes of a particular container profile can be shelved there.
- The ability to relate locations to other locations or 'sublocations'. This would support auto-completion of bay, row, floor, room, building information when a box is associated with a shelf (e.g. barcode is scanned). It looks like the original data model for ArchivesSpace called for locations to be able to be related to other locations

[1] It would be good to confirm these and discuss these data model ideas with ArchivesSpace developers as the latest data model we could find for ArchivesSpace is dated 2011: http://archivesspace.org/sites/default/files/drupal7/draft-ERD-2011-11-07.pdf and is high-level.

It is certainly possible to model intellectual entities, expressions, and manifestations / items as related archival objects and associate them with locations. Clarifying how we will use archival objects to model these relationships when needed can be done via good descriptive policy and, ideally, by the templating functionality described in a user story in the ArchivesSpace application roadmap. There are ways in which the current application is confusing (e.g. the term 'instances' could be confusing as it is purely about physical location assignment, as is the option to assign a 'Type' to instances) and could be improved but they are slightly outside of scope of location, barcode and container management.

Testing results summary - Rachel

The requirements were then turned into a testing script by devising specific real-world examples that are routinely carried out by archivists, librarians, and other associated staff while using the application. These actions were performed in the test instance environment and given a determination of "Supported," "Not Supported," or "Inconclusive." Some functional requirements were tested using multiple different examples to account for the myriad ways in which this function might be used by the different repositories or types of users. Requirements identified as "inconclusive" were either partially supported or did not perform as was expected (that may indicate a bug rather than a gap in functionality).

In addition to the tests performed, existing technical documentation (such as the User Manual and technical specification documents for different modules of the application), features and improvements incorporated into release v1.3.0, and the "Open" or "In Progress" stories in the ArchivesSpace JIRA site were consulted to confirm what functionality is planned to be added to the application around location, barcode and container management. None of the fixed bugs or the new features announced in release v1.3.0 significantly change the outcome of the testing results (one feature will update all reports to make them repository specific, but this would still require some degree of configuration or manipulation on our part). Of the 72 open stories (there are currently no stories designated "In Progress"), three have implications for our location, container and barcode management needs, although the statements are so general at this stage that it is difficult to know if they will solve specific issues that were identified as being "Not Supported." Multiple open stories refer to improvements to import functionality, which may tangentially impact location, container, and barcode management. The following open JIRA stories that have been identified as particularly relevant are:

Story Number	Description
AR-1250	Update all reports using container information
AR-1246	Update exporters to work with new container model
AR-1245	Update importers to work with new container model

Summary of Aeon integration - Sally

As we are beginning Aeon implementation in Winter 2015, it is timely to consider how location, barcode and container information may be exchanged between ArchivesSpace and Aeon and specifically whether there are any fields or functionality in ArchivesSpace that we will need to support Aeon functions.

Transferring information from ArchivesSpace to Aeon via EAD

To print call slips for all three reading rooms from Aeon, we will need several pieces of information that Maureen Callahan identified in an element-by-element evaluation of TamWag's DACS compliance - these are:

- Reference code element / call number - <unitid>
- Repository note <repository> - controlled value for routing
- Collection title <unittitle>
- Container information <container> - call number + box number + barcode
- Physical location note <physloc> - noting whether material is offsite or onsite (nice to have)
- Access restrictions <accessrestrict>
- Location - nice to have - wing / room / bay etc.

The important thing to consider here is whether ArchivesSpace supports capture of all of this information and can output them in EAD files. ArchivesSpace does support capture of all of these fields and supports controlled values for repository code, which will be used by the application to determine which queue a request goes into and therefore must be regular.

The EADs ArchivesSpace generates includes all of this information except for, understandably given security issues, location data. If we wanted to automatically include location data on a call slip, a script would need to be developed that called the ArchivesSpace API or database and pulled through location information into Aeon. This is a 'nice to have requirement' and the important information to note at this stage is that ArchivesSpace allows for capture of the information that would enable this functionality.

As part of the Aeon implementation deliverable, we will test the EADs produced by ArchivesSpace with Aeon and may need to conduct some data remediation.

Transferring information from ArchivesSpace to Aeon via ArchivesSpace public interface

If we decide to use the ArchivesSpace public interface as our special collections discovery system, it would be beneficial if ArchivesSpace implemented a 'cart' or 'bookbag' feature and pop-up interface to allow entry of request data and then formed an Aeon-friendly request to pass that data to Aeon. The ArchivesSpace Program Manager agreed to include this integration functionality in the core ArchivesSpace development queue at a recent Aeon Users' Group meeting.

Syncing temporary locations in Aeon with ArchivesSpace

It would be nice but not essential if temporary location data could be dynamically updated or 'synchronized' back to ArchivesSpace when changed in Aeon. There is community discussion underway about how this may work. This is not a critical requirement for us, and therefore we do not propose paying for this integration piece but we will monitor any community work underway to develop this.