

Bound-with data model

This model proposes to overlay the existing modeling of relationships between Instances, holdings records and Items with a new relationship table that ties multiple holdings records – and thus multiple Instances – to one circulated Item, the bound-with.

The proposal aims to avoid breaking compatibility changes and to minimize the need for adaption of business logic in other components that operates on Item data, most notably Circulation and to some extent ui-inventory and Inventory search. It does that at the cost of introducing a certain element of redundancy.

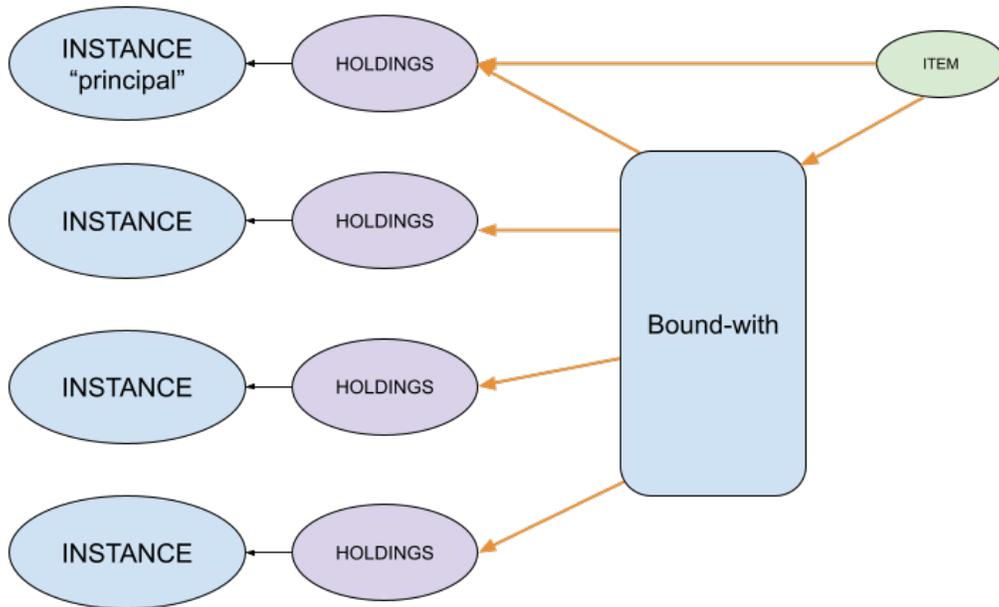
Background and motivation

The current data structures consists of Instances, holdings records and Items such that one Instance has zero or more holdings records, each with zero or more items. The item is the unit of circulation and thus holds an item barcode as well as the circulation status of the item. The Instance describes the title with meta data. The Item is thus related to one and only one holdings record and in turn to one and only one Instance.

The current model does not account for the bound-with scenario where you still want one unit of circulation but want to describe it in multiple meta data records (Instances) because the bound-with is a compilation of multiple distinct pieces of work.

Proposed model

Bound-with. Single Item linked to multiple Instances



The proposed model suggests to retain the model of one item being linked to one holdings record, being linked to one instance – all currently enforced by database level constraints – and the Item of this relation will continue to be the unit of circulation.

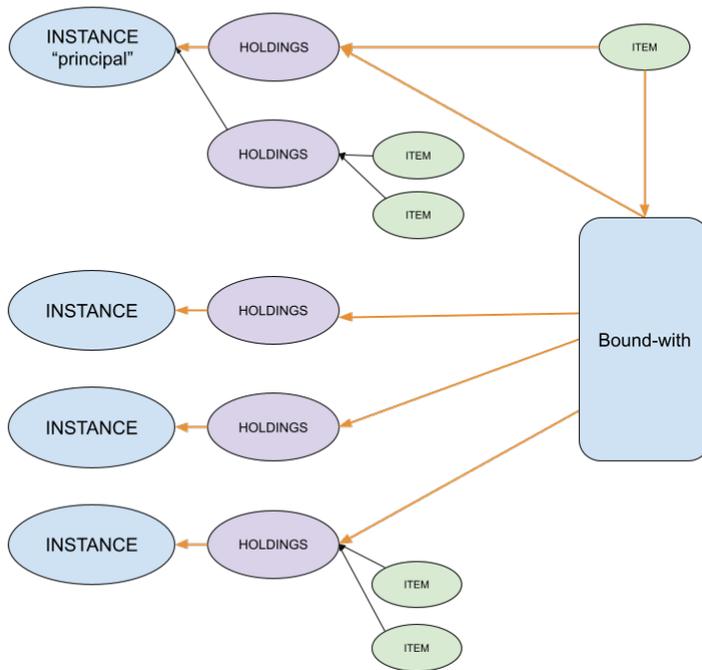
However, to model that multiple Instances are included in the circulated Item, a separate table is created, which would hold the relationships between the item / unit of circulation and all the holdings that are included in the bound-with.

The redundancy mentioned comes from the fact that one of the holdings records are linked to the Item in two ways as illustrated in the diagram.

While this is not the cleanest possible data structure it is at least warranted to have one holdings record -- and thus one Instance – being the principal entity of the bound-with, since we have to pick a title for display wherever the bound-with appears, for example in the Inventory UI or the Checkout UI as "Some selected title [and other titles]". It's not feasible to display some sort of concatenation of all the sub-titles of the bound-with wherever the bound-with appears in the UIs.

Supporting non-bound-with copies of the same titles alongside bound-withs

Bound-with and non-bound-with copies of the same titles



What stays the same

The unit of circulation is still the one item.

The Item will still be directly linked to one holdings records by its mandatory `Item.holdingsRecordId` property.

The `barcode` will still exist on this Item and only on that.

A UI Inventory search of Instances by `barcode` will thus still hit that one item and bring up the principal Instance.

The Instance details page would still display the holdings records of the Instance and the Items directly linked to those holdings records.

What changes

The Instance results page should have a flag on any Instance that is part of a bound-with and display a title similar to "A title [and other titles]" in the list (more on how to do this later)

When the Instance is part of a bound-with, the Instance details page should pick up all the other holdings records from the bound-with table and display them in some fashion on the details page with all their associated titles that make up the bound-with – for example in a bound-with accordion.

In other places where the bound-with item occurs – for example in checkout – the title should similarly be presented with those indications that it is a bound-with.

What could also change

It would require changes to the search by `barcode` to have it pick up all the other Instances of a bound-with but this is not deemed crucial as long as the bound-with flag is displayed on the principal Instance in the results list and the other Instances are displayed in the Instance details page.

How to flag that an Instance is part of a bound-with, or that an Item *is* a bound-with.

The Instance would be known to be part of a bound-with if it has a holdings record that appears in the new bound-with table. However, it is non-trivial for a client, specifically for `ui-inventory`, to resolve this per Instance, so the suggestion is to have the business logic back-end module, `mod-inventory` perform that look-up whether for a single Instance look-up or across an entire result set of Instances, before returning the Instance responses to the client. For the client the bound-with indicator would appear as a boolean property `"isBoundWith": true`.

The Item record should have a similar flag set coming through `mod-inventory`. Holdings records are not passed through `mod-inventory` at this point. As it happens, `ui-inventory` does not need the flag according to current drafts for the UX for the display of the bound-with aspects.

Back-end API for retrieving a composite bound-with record set

Eventually the UI should display all titles and holdings in a bound-with item. I would thus likely be convenient with an API in mod-inventory that could build such a response from the raw bound-with structure in mod-inventory-storage. The list would at minimum contain instance ID, title, and holdings ID for each piece in the bound-with.

Edge case. A given holdings has multiple bound-with copies.

Use case: Quarterly magazine, acquired in three copies, two of which are eventually bound into items of similar content – two bound-withs - where one is containing four issues of the year, and the other is containing three issues of the year.

Inventory storage allows for this model – by allowing that multiple bound-with items can exist through a relationship with one and the same holdings-record.

Quarterly magazine, acquired in three copies, two of which are eventually bound into items of similar content – two bound-withs (one containing four issues of the year, and another containing three issues)

