

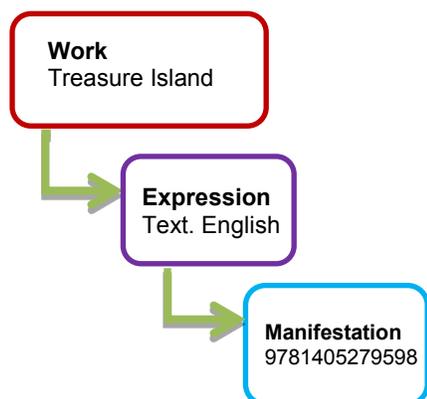


BDS is a cataloguing agency, creating bibliographic records to library standards. We have built our database over the past twenty-one years using a data entry system that has received regular upgrades to keep abreast of both technological and bibliographical developments. With its success based upon the quality of its data, BDS always strives to use the most efficient and effective methods for data entry and the recent developments around RDA and linked data persuaded the BDS development team that it was time for a replacement. Our long-standing and highly

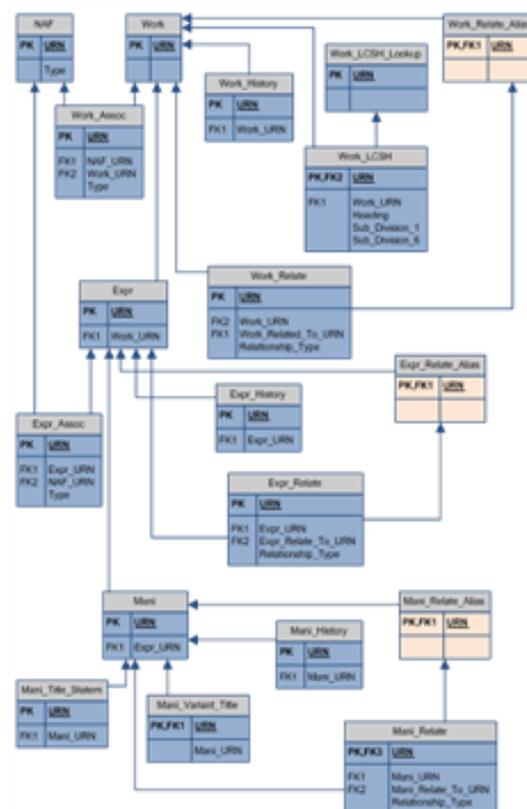
successful system has its roots in DOS, and uses a programming language which is increasingly difficult to support. It requires training in MARC21 and internal processes, as well as in the standards that determine the content of the bibliographic record. Here, I shall describe the process of designing new software for a cataloguing and associated data entry system which, after a staff competition to name it, is called "Aura".



BDS implemented Resource Description and Access as the cataloguing standard in January 2013. RDA is based on the conceptual model FRBR, using concepts of works, expressions and manifestations. A work is realised through its expression, which is embodied in the manifestation. It became obvious to us, and the rest of the cataloguing world, that RDA does not fit well in MARC. In some ways this made implementation easier, as it meant smaller changes in practice, but MARC's flat-file structure doesn't allow us to fully realise the relationships involved. It is manifestations which we catalogue, and to date we've created an individual, isolated record for each manifestation.

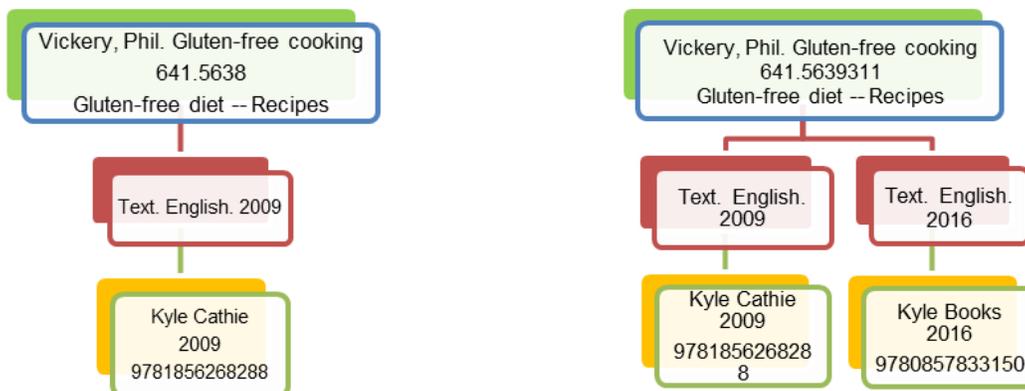


The FRBR model underlying RDA, the advent of linked library data, and the idea of a 'master' record that we'd had for a while, all combined to suggest a relational database structure for the new data entry system. A relational database holds data in tables which have relationships to one another. Note the use of 'data' rather than 'records', which is a significant change of mind-set. The 'cluster' of associated data for a manifestation could be called a record, but we're going to have to find ways to think differently about bibliographic information.



The relational database is powerful, because if we change data in one table, all connected data is also affected. For example, when the first manifestation of Phil Vickery's Gluten-free cooking was catalogued in 2009, there was no specific Dewey number for gluten-free cooking. By the time the 2016 updated edition was published, we did have a specific Dewey number for gluten-free cooking. By updating the Dewey of the work, we would update the Dewey for every manifestation of that work.

It's also possible for BDS to have records for multiple manifestations of a work, but only catalogue one of them book-in-hand. The remaining manifestation records are currently based on the publisher's information at the CIP stage. With a joined up database, any confirmed details which affect the work and expression for the manifestation we have in hand will then also positively affect the data for the other manifestations.



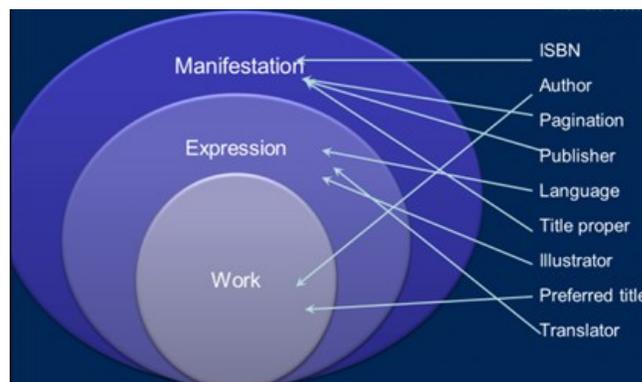
Whilst many manifestations are part of a network of connected resources, other publications stand alone and so there's no existing data to connect to. This has to be borne in mind to make sure the new system is still efficient when all of the data has to be created from scratch.

So, we decided that the new database and interface should:

- use the FRBR model
- be able to produce MARC records
- bear in mind developments for post-MARC bibliographic data
- support rather than hinder productivity
- be able to import data, both MARC data from the existing database and ONIX data from publishers.

The data had to be as granular as possible, and the interface had to have as much of our cataloguing expertise built in as we could write logic statements for. The work is far from over, but there have been three main areas of effort: the model for the database, the 'translations' to import and export MARC records, and the business rules which support the creation of good data. I'll outline the kinds of issues involved with each of those.

We began by listing all the bits of information we wanted to include, and deciding which table they belonged to. In addition to the RDA elements, we needed to capture additional information that the MARC record would need, and BDS requirements such as trade edition statements. An easy, early decision was to use as many identifiers as possible. Each entity having a unique identifier is inherent in the relational database structure, but we also want to store persistent global identifiers like ISBNs, and NACO control numbers. This supports future linked data developments, whether using Bibframe or other standards.



The RDA implementation of the FRBR model is only dealing with descriptive cataloguing, and not with classification and indexing. There were problems to solve where that model interfaced with other knowledge organisation systems. BDS uses Library of Congress Subject Headings (LCSH) for subject indexing, and we saw the subject of a resource as being at the work level, but we couldn't simply assign all LCSH to the work level in the model. This is because LCSH include form headings such as "Talking books" and "Large type books" which pertain to expression and manifestation levels respectively. Our solution was to allow expression and manifestation level LCSH.

We also realised we had an issue with aggregate works. Aggregate works are anything that can be split into separate works, so a book containing articles by different authors is the obvious example. There are many other examples, though: omnibuses of novels, continuing resources, and resources which include the same work in different expressions. When one thinks about each work included in the aggregated work, and all the potential relationships between those works, the model gets complicated. The principal problem we had to solve was self-referencing works, i.e. a publication which both *is* and *is about* a particular work, such as the text of Hamlet plus study notes, or a Bible which has commentary as well as the Bible text. Our pragmatic solution is to create works where we need them, and avoid self-referencing.

After resolving modelling issues, the next step was to tackle the interface with MARC. Importing data that's less granular than your database is problematic; exporting requires a high level of complexity, but it does fall to logic - so long as the database is granular enough.

The 245 Title field and the 300 Physical description fields are both examples of difficult areas in MARC, because the subfields contain more than one kind of data. A human looking at a 245 field can determine which full stops are there for abbreviations, and which are there for ISBD punctuation, but a machine cannot:

Incoming MARC	Aura database	Outgoing MARC
245 04 \$a The history of the world. \$n Volume 2, \$p Kingdoms and empires	Title=The history of the world Filing characters=4 Part number=Volume 2 Part title= Kingdoms and empires	245 04 \$a The history of the world. \$n Volume 2, \$p Kingdoms and empires.
245 04 \$a The history of the world / \$c edited by J. Smith. Volume 1, The first civilizations.	Title=The history of the world Filing characters=4 Statement of responsibility=edited by J. Smith. Volume 1, The first civilizations.	
	Title=The history of the world Filing characters=4 Statement of responsibility=edited by J. Smith Part number=Volume 1 Part title=The first civilizations	245 04 \$a The history of the world / \$c edited by J. Smith. Volume 1, The first civilizations.

In the 300 field, the same fields are used differently for different kinds of material:

300 \$a 9 CDs : \$b digital, stereo
300 \$a 456 pages : \$b illustrations (black and white) ; \$c 24 cm

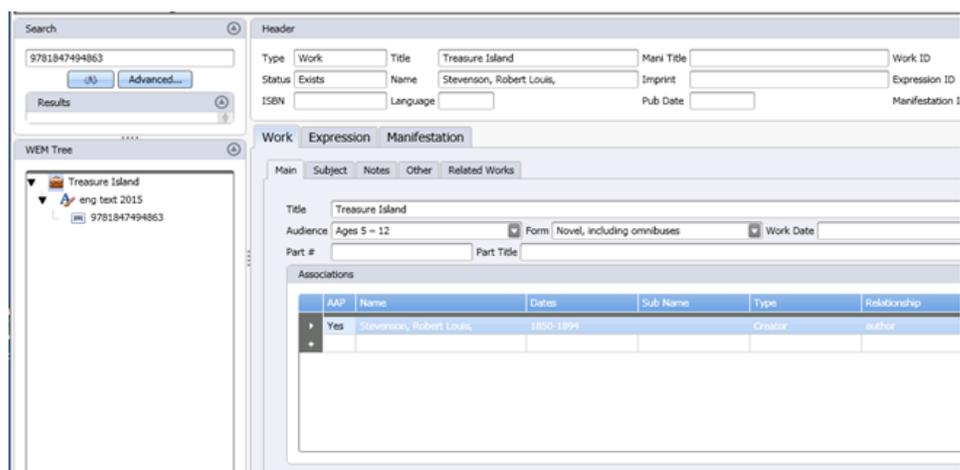
Both of these fields require a lot of complicated programming when importing, and a lot of information captured specifically to build the MARC for export, such as detailed labelling of the Title statement. An icon produces a MARC view of the record in "Aura", allowing users to check the output.

Last but not least is the business rules, which in this context means the kind of rule that says "If you've identified this resource as having the Main content type of 'text', then you cannot have data in the field for Expression duration, because that only applies to Spoken word, Performed Music, or Moving images." As you can imagine there are a lot of logic statements which could be deployed, and we're still adding to them.

There are some basic defaults, which 9 times out of 10 will save a few keystrokes. For example, the language of the expression defaults to English. Then there are conditional defaults, for example if the work form chosen is Music then the main content type default changes from text to notated music.

The final component to the business rules is error checking – a work has to have a title, and other core pieces of information, so the system won't let you save it unless the core requirements are met. If the physical format chosen is incompatible with other data elements, the system can flag that up to the user.

The user interface still needs a lot of work and testing, but a significant change for us is the use of a tabbed layout. Since we're not constrained by MARC order, a tabbed layout optimises the visibility of data elements and their accessibility for entering and editing by the user. There's a Work, Expression, and Manifestation tab, and each of those has tabs for specific aspects of the data.



There's a search box, and double clicking on a result brings up the relevant information in the main screen, and a WEM tree in the bottom left.

If I search for an ISBN, and it's not in the database, the system looks for it in the old database and the publisher's ONIX database. When a record is found, the user then contributes to how it's imported. The system attempts to match to works and expressions already in the "Aura" database, and the user can accept the suggested matching work, or choose a different one, or demand a new work is created. If an existing work is selected, the system goes on to look for matching expressions. If you decide it's a new work, then obviously it will need to be a new expression too. If matching works/expressions have been found, then we have far less to do than creating the record from scratch.

A significant training issue for this new system will be to ensure users realise the effect that changing information in just one field can have. Depending on where a change is made, a cascade of differences could result. There will therefore be different levels of user status, with only the most skilled and trusted being able to make high-impact changes. There are also decisions to make about which field changes are worth triggering a "resend record" status for. For example, updating the Dewey of a work is useful to virtual retrieval, but is not going to change the spine labels on items on library shelves.

If cataloguing becomes more about identifying entities and their relationships and we move from text based bibliographic records, to use as many identifiers and discrete pieces of data as we can, I think cataloguing will be a more intellectually satisfying process. It will also facilitate better search and retrieval for users. Creating "Aura" continues to be an interesting and exciting project where BDS, in line with its ongoing philosophy of quality and accuracy dedicated to accommodating the future, is leading the way in its field. Now, we are looking forward to getting to the testing stage and further refining "Aura".