RIOXX as an effective mechanism for the efficient exposure and aggregation of repository content

George Macgregor, University of Strathclyde
@g3om4c

Those who have been active in repository-land since the emergence of the post-Finch Open Access policy of UKRI (then known as RCUK) will probably be familiar with the RIOXX metadata application profile. RIOXX was originally developed by Paul Walk (UKOLN and then EDINA) alongside Sheridan Brown (Repository Consultant) and was sponsored by RCUK and endorsed by HEFCE. Versions 1.0 (2013) and 2.0 (2015) of RIOXX were largely concerned with describing research outputs that were the outcome of RCUK funding and therefore within the scope of RCUK's new Open Access policy. Describing and exposing data about these outputs would not only enable new opportunities for their discovery and aggregation via OAI-PMH, but would also allow RCUK to monitor aspects of policy compliance among institutions.

Despite being more widely adopted than some of you might imagine (64 UK repositories the last time I checked), universal adoption of the profile was slower than anticipated. A reason for this slow adoption was the ambiguity surrounding its use and status. RIOXX was ostensibly created to aid monitoring of RCUK OA policy implementation; but RCUK chose not to mandate its use, nor were institutions compelled to use it in their OA reporting. RCUK failed to create an aggregation of RCUK funded research content or endorse an existing solution, in the way e.g. the European Commission did with OpenAIRE. The perceived lack of a compelling use case stymied uptake of what was a superior metadata profile for repositories. Unfortunately RIOXX, as it was originally conceived, became orphaned from its original raison d'etre when RCUK and HEFCE became UKRI– that is, until now. Indeed, this 'perceived' absence of a use case was, and is, just that: perception.

Since autumn 2019 UKCORR has adopted responsibility for RIOXX, continuing the development and governance of RIOXX through a recently formed 'RIOXX Governance Group' (RGG). The RGG has been busy reviewing RIOXX, gathering requirements for revision and is currently in the process of drafting version 3.0 for public comment -- because it transpires that RIOXX demonstrates some best practice in metadata modelling and exposure. In this context, RIOXX is not primarily and only about UKRI compliance, or compliance with the REF 2021 Open Access Policy – it is about better exposing repository data for discovery, re-use and aggregation. And this, I think we can all agree, is what open repositories are really all about.

Why does RIOXX remain relevant?

But why is RIOXX superior? The issue relates to how most repositories and institutions have implemented OAI-PMH over the years. If we venture back into the mists of time, the OAI-PMH standard documentation describes how Dublin Core (DC) was to be a minimum metadata expectation for repositories. In fact, at that time (circa early noughties) there was an unspoken assumption among digital library and repository developers that most OAI-PMH implementations would offer something more than DC; that richer metadata, demonstrating greater specificity, would be provided in addition to OAI_DC. In other words, OAI_DC would be a simple fall-back standard for baseline interoperability and not the principal metadata exposed by repositories. However, the year

This article was published as an invited blog post on the UKCORR blog: Macgregor, G. (2020, Aug 25). RIOXX as an effective mechanism for the efficient exposure and aggregation of repository content. UKCORR. Available: https://www.ukcorr.org/2020/08/25/rioxx-as-an-effective-mechanism-for-the-efficient-exposure-and-aggregation-of-repository-content/

is 2020 and it is now fair to state that this promised land of rich, beautiful metadata never happened. Instead the 'minimum requirement' became the default metadata offered by the OAI-PMH interfaces of repositories. The inadequacy of OAI_DC has been demonstrated by the various metadata application profiles which have emerged over the years. These profiles often attempted to compensate for the inadequacies of OAI_DC which, on its own, has been insufficient to service the requirements of repositories and aggregators. In fact, the lack the granularity and specificity in OAI_DC is such that many services now simply refuse to use.

These limitations of OAI_DC have not gone away, and if repositories wish to expose their content to the widest possible audience and observe their data and content being used to facilitate open knowledge creation, they need to be better at describing their content. This means using something other than OAI_DC and, in this case, preferably RIOXX.

RIOXX provides a number of benefits which will be described elsewhere in future; but perhaps the issue that crystallizes the problem with OAI_DC most is the disparate way in which `dc:identifier` is implemented within repositories. The broad scope of this element in simple DC has resulted in its inconsistent interpretation. Irrespective of which repositories have interpreted `dc:identifier` correctly, the current situation is especially problematic for systems that wish to harvest, aggregate repository and re-use repository data, such as CORE (core.ac.uk). Such systems do not know whether the content of `dc:identifier` is linking to another identifier or the described content -- and they are insufficiently informed about the digital resource they are seeking to make assumptions about the format of the resource(s), the number of files, the availability or visibility of the resource(s), and so forth. A more detailed explanation of this particular problem can be found in an Open Repositories Conference paper by Petr Knoth.

Determining answers to these questions can make the process of harvesting and content aggregation unnecessarily complex. Such systems may need to revisit the same item within a repository dozens, perhaps even hundreds – sometimes thousands, of times in order to successfully interpret the content of `dc:identifier`. Not only does this create unnecessary computational load for the harvesting system, it also places unnecessary load on individual repositories too since they themselves need to be crawled dozens, perhaps hundreds – sometimes thousands, of times by harvesters. Harvesters may also find themselves forced to devise repository specific techniques in order to crawl a repository satisfactorily. All of this is unsatisfactory.

However, for systems like CORE – which uses RIOXX for harvesting wherever a repository makes it available – the conundrum of `dc:identifier` is addressed. In RIOXX the content scope of `dc:identifier` has been narrowed to only permit an HTTP(S) URI pointing to the resource being described, enabling greater efficacy in harvesting and aggregation, and a reduced load on individual repositories. For example, from version 2.0 of the RIOXX specification:

> *This field [dc:identifier] MUST contain an HTTP URI which is a persistent identifier for the resource. The purpose of this field is, through direct identification of the resource, to allow access to it, therefore it is RECOMMENDED that this identifier should point to the actual resource being described by the RIOXX record (typically a file in MS Word or PDF format), rather than to an intermediary resource such as a repository web page…*

By narrowing the scope of `dc:identifier` RIOXX immediately solves an issue that has been plaguing aggregators and OAI_DC repositories they attempt to harvest.

This is but one area, albeit an important one, where adoption of RIOXX can improve the quality and efficacy of harvesting and ergo the re-use potential of the data.

RIOXX futures…

UKCORR and the RGG are committed to maintaining and developing RIOXX in a platform agnostic way. This means maintaining the profile in the interests of the UK repository community as a whole, UKCORR members, and for open research more generally. Although there are significant differences between version 1.0 and 2.0 of RIOXX, both are essentially application profiles of DC. This will also be true of version 3.0 when it arrives which, the RGG hopes, will be in the autumn of 2020. The RGG is also committed to being 'aggressively open' in its management of RIOXX, so expect version 3.0 to be announced through all the usual channels soon. Interested readers can also visit the RIOXX GitHub repository to explore proposed revisions and read the minutes of RGG meetings.