

Using OAI-ORE to Transform Digital Repositories into Interoperable Storage and Services Applications

*In the digital age libraries are required to manage large numbers of diverse objects. One advantage of digital objects over fixed physical objects is the flexibility of 'binding' them into publications or other useful aggregated intellectual entities while retaining the ability to reuse them independently in other contexts. An emerging framework for managing flexible aggregations of digital objects is provided by the Open Archives Initiative (OAI) with its work on Object Reuse and Exchange (ORE). This paper will show how OAI-ORE is being used to manage content in digital repositories, in particular institutional repositories, and has the potential ultimately to transform the conception of digital repositories.*

By David Tarrant, Ben O'Steen, Tim Brody, Steve Hitchcock, Neil Jefferies and Leslie Carr

Introduction

One of the main decisions to be made when starting a digital repository is which software to use. Currently there are a wide range of choices, including open source and proprietary systems. Most of these choices are in active development and some have been available for nearly a decade, which represents a high degree of stability in the relatively young history of digital repositories. Not every institution creating a repository will make the right initial choice of software, however, and over time the needs of a repository may evolve. Not every repository software will survive. The well managed repository will have a preservation strategy, but what happens if the underlying software has to change? While repositories, especially the open source variety, are not designed for data lock-in, different data models mean that lossless data transfer between repositories has not been simple. This is about to change with the emergence of OAI-ORE.

The starting point for our work was an investigation into institutional repository (IR) preservation services by the JISC-funded Preserv 2 project [25], which wanted to find a way of effectively replicating a whole IR across any repository platform. The aim was to avoid tying digital objects to the repository software used to manage them. The vision was of a set of low-level resources, which are then managed by a series of services including repository software such as EPrints and Fedora.

The OAI-ORE specification includes approaches for representing digital objects and facilitates access and ingest of these representations beyond the borders of hosting repositories, enabling a new generation of cross-repository services. In this way OAI-ORE standardises the description of the relationship between digital objects. This relationship could be between versions of an object, such as might be found in a repository record, or aggregations of objects, such as a Web page with images, or a collection of chapters in a book. We also envisaged applying OAI-ORE to a much larger collection – a complete repository.

This paper describes an award-winning [27] rapid development project, entered into a competition [23] sponsored by the Common Repository Interface Group (CRIG) [5] at the Open Repositories Conference 2008 [21]. Using OAI-ORE with a simple extension that enables the import process to operate more quickly, this project was able to replicate not only objects but also all the metadata and history data for those objects across repositories. The demonstration combined OAI-ORE with the Fedora and EPrints repository platforms and culminated in a lossless transfer of two live archives from one software to the other.

The implication of this approach is that repositories could become interfaces to remote storage where content is represented and accessed via an OAI-ORE resource map. Rather than repositories storing and organizing objects internally and then revealing this content to other services through OAI-ORE maps, the repository understands OAI-ORE maps at the lowest level. Further, these maps need not be exclusively managed by the repository itself. For example, we might have resources (documents, pdfs) stored directly in an open storage platform and an OAI-ORE map binds these resources into collections. In this scenario, objects and resource maps are no longer controlled by, and reliant on, a single repository software. Instead, we can envisage the prospect of many repository softwares (EPrints, Fedora, etc.) running over one set of resources. While this vision is some way off, the application of OAI-ORE makes this transformation from repositories-as-software to a services-based conception a real possibility.

This paper provides a concise introduction to the OAI-ORE specification, from its roots in Web architecture through the development of multiple serialisation formats. We focus on how OAI-ORE can be used to represent aggregated publication records, expanding this to show how OAI-ORE can be used to map an entire repository and its content. We explain how OAI-ORE was used to transport the entire contents of one repository into another, and vice-versa (with the two repositories running seemingly incompatible softwares). Since the initial demonstration (done one day before OAI-ORE was formally announced [19]), some of the interfaces that were developed have been integrated into the respective repository softwares, showing that OAI-ORE already provides real benefit to the digital repository community.

We also look briefly at possibilities for further work in this area. OAI-ORE is still in its infancy and currently there are not many user-facing applications. We expect that this will change and that OAI-ORE will play a greater role in binding digital objects both on the desktop and on the web without needing to move those objects into one static location. Using OAI-ORE at a low level has a number of implications for repositories, in particular being able to construct a repository which consists of a set of interchangeable services. To understand how repositories might change, first we need to consider the current purpose and structure of repositories, in particular, of IRS.

Institutional repositories and preservation

IRs are designed to host and disseminate academic research and institutional output, and they play a key role as a platform for managing digital content. A good IR will be driven by the needs of the institution, and the main benefits will arise from being embedded in institutional processes and cultures. These benefits include growth of content accessible to all Web users, integration into institutional web presence, and automatic profile and CV generation for researchers. IRs have also been shown to aid academic research evaluation [4].

Each repository software solution performs three basic functions: ingest, management and dissemination of content. Although the various repository software solutions are solving the same, or similar, set of problems, each platform is constructed with differing low-level architectures, including fundamental differences in how files and metadata are stored and organised. These differences mean that you cannot simply transport the raw contents from one platform to the other.

From a high level standpoint, the repository platforms strive to provide compatibility by supporting protocols and specifications which allow easy ingest and discovery of the data they contain. Figure 1 models a simplified three-stage repository showing example interfaces, or plug-ins, which can be used at each stage. An ideal repository implementation will take a completely modular approach such that plug-ins and extensions can be developed for each stage of the model. In this paper, the import (Get Content) and export (Serve Content) plug-in layers provide the mechanism for interoperability with other systems.

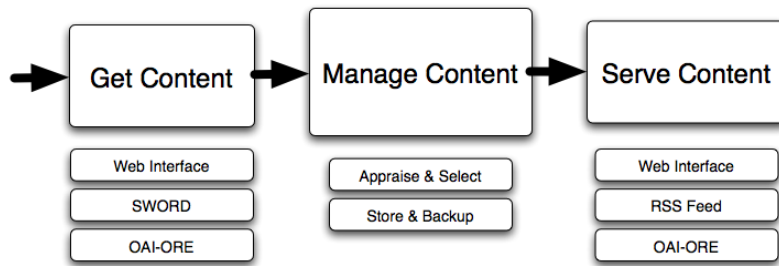


Figure 1: The 3 Stage Repository Model.

While OAI-PMH has been fundamental in enabling interoperability between repositories, it is designed as a method of discovery; there is no corresponding import protocol with the same application programming interface (API). Until now there has been no standard way of abstracting a complete object such that it can be transported from one repository platform to another without some loss of data. This is where OAI-ORE comes in. OAI-ORE specifies import and export interfaces to enable to re-use and exchange of digital objects. From a digital preservation perspective, this enables future migration of objects to a newer platform while preserving the functionality expected from the digital repository.

A quick overview of OAI-ORE

'Open Archives Initiative Object Reuse and Exchange (OAI-ORE) defines standards for the description and exchange of aggregations of Web resources. These aggregations, sometimes called compound digital objects, may combine distributed resources with multiple media types including text, images, data, and video. The goal of these standards is to expose the rich content in these aggregations to applications that support authoring, deposit, exchange, visualization, reuse, and preservation. Although a motivating use case for the work is the changing nature of scholarship and scholarly communication, and the need for cyberinfrastructure to support that scholarship, the intent of the effort is to develop standards that generalize across all web-based information including the increasing popular social networks of "web 2.0."' [20]

This section looks at the relationship between OAI-ORE and IRs and how OAI-ORE can be used to bind documents (aggregating metadata, history, content) in a repository as well as to bind the repository itself (aggregating objects and repository configuration). First, we consider the technical model behind OAI-ORE and how it has been constructed.

The OAI-ORE model

OAI-ORE is based entirely on the architecture of the Web (as outlined in [11]) and advocates use of recent developments in the areas of the Semantic Web, Linked Data and Cool URIs [3]. The biggest influence on OAI-ORE, however, is the RDF model (described best by the W3C RDF Primer [16]), which uses the idea of triples to describe things. A triple comprises a subject-predicate-object statement where each part is a URI, with the exception of the object, which can also be a plain text value. Figure 2 presents a simple repository-focused example of the RDF model, using the Dublin Core namespace [5] to provide all of the predicates (and shows the classic use of the RDF model to annotate objects with metadata).

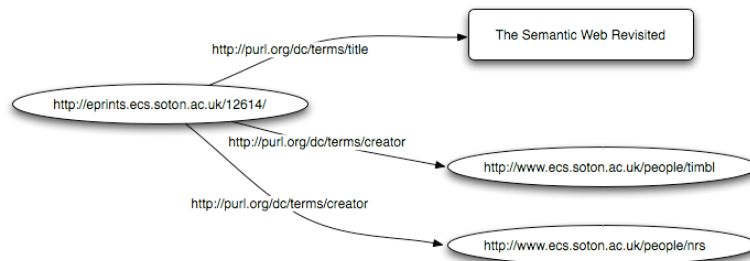


Figure 2: An RDF representation of publication metadata.

OAI-ORE focuses on objects and the relationships between these objects, leading to a use similar to that shown in Figure 3. Here our publication object now has two parts, the publication itself (pdf) and the xml record which in this case represents the stored metadata.

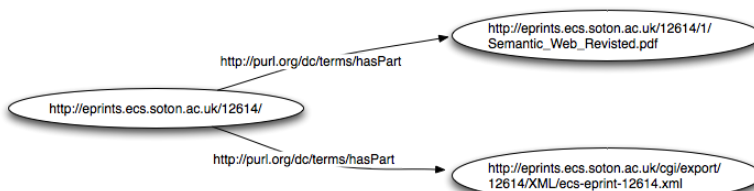


Figure 3: RDF representation of a publication using Dublin Core.

## Aggregations

On top of the RDF model, OAI-ORE introduces the concept of Aggregations and Aggregated Resources. An Aggregation is simply a set of Aggregated Resources, all of which are represented by URIs. Figure 4 is adapted from the ORE Primer [13] and demonstrates how this concept applies to a publication resource in a repository. Now that we are using the OAI-ORE methodology we see the use of the "ore:" namespace (shorthand from "http://www.openarchives.org/ore/terms/" to "ore:"), which is designed specifically for OAI-ORE.

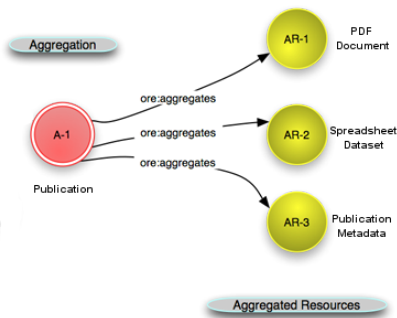


Figure 4: Example OAI-ORE Aggregation of a Publication.

While Figure 4 shows a single publication record, the abstract concept introduced by OAI-ORE is much more powerful and allows nesting of Aggregations. For example, the highest level Aggregation could be the repository and the Aggregated Resources thus become the publications, which in turn contain their own Aggregated Resources. There is no limit to the depth with which resources can be aggregated, although it is not recommended to go to too many levels due to the recursive operations that will need to be performed to import these resources.

## Resource maps

Resource Maps represent the highest level of the OAI-ORE model. A Resource Map has a URI and is used to describe a single Aggregation. Note that a Resource Map can only link to a single Aggregation in the OAI-ORE model. Figure 5 (taken from the ORE Primer [13]) shows the Resource Map (ReM-1) to the left of the Aggregation with some example metadata that is expressed using the Dublin Core namespace. Based on RDF, the Resource Map is able to support any namespace and metadata the user wishes to add.

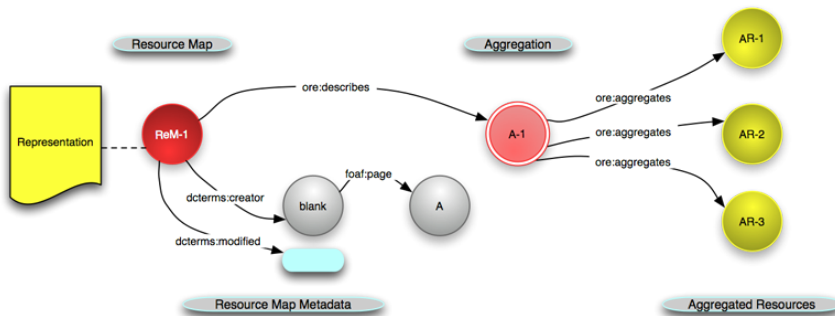


Figure 5: Complete OAI-ORE Model

## Representations

RDF/XML is the obvious and most flexible model for representing OAI-ORE Resource Maps, but it is not the only one. Another is the popular Atom/XML serialisation [10], which is in widespread use in Web 2.0 services. Atom is supported by many browsers that are able to display an Atom document in human readable form. Atom is also currently supported by Google as a means to update its search engine. Table 1 links to an example of each type of serialisation, both of which represent the same publication taken from an EPrints archive enabled with OAI-ORE functionality.

| RDF/XML Serialisation<br>Example in RDF/XML | Atom/XML Serialisation<br>Example in Atom/XML |
|---|---|
| <a href="#">[Link]</a>                      | <a href="#">[Link]</a>                        |

Table 1: Example OAI-ORE Resource Map serialisations of the same publication.

This section has covered the basics of OAI-ORE and outlined the parts which directly affect our work. For a more in depth explanation and additional feature specification we recommend the OAI-ORE home page [20]. It provides links to further serialisation specifications and the ORE Primer [13], much of which we have summarised here. More information on the RDF model can be found through the W3C's RDF Primer [16].

## Combining OAI-ORE with the repository

In this section we demonstrate how OAI-ORE can be used to transfer resources from one repository software to another, as performed for the award-winning demo at Open Repositories 2008 [27]. The full demonstration required writing export plug-ins to reveal publication records as OAI-ORE resource maps and corresponding import plug-ins for each repository software platform. As expected, the export plug-ins were the easiest to create as all of the data exists already and the IR model is a good fit to the OAI-ORE specification. Building the import plug-ins led to the discovery of an issue best solved in the OAI-ORE specification rather than on a per-repository basis. We discuss this issue and solution further in the "Extensions to the OAI-ORE Specification" section.

By using EPrints and Fedora, both well established repository software platforms, as the basis for the development and trial of the OAI-ORE plug-ins, we had the advantage of two platforms that can be readily extended both in terms of import and export. Our aim was to see if OAI-ORE could indeed enable the re-use and exchange of repository objects.

### The demo

To demonstrate the power and ease of applying OAI-ORE to a real world scenario, two real world repositories were chosen as data sources. The example Fedora repository was the Oxford Research Archive [24]. The data from this archive would be harvested into an empty EPrints repository configured with the import plug-in. The chosen EPrints repository was that of the Open Repositories Conference itself [22]. The harvesting would also be performed in the opposite direction, transferring objects from the EPrints repository to an empty Fedora repository. With the four repositories configured (2 originals used for export and 2 temporary ones used to import data from the opposite software), a small script was run on each destination repository pointing it at the resources to import, at which point the process was automatic. A video is available showing the demo and questions from the competition judging panel [17].

### Building the plug-ins

For the purposes of completing the demonstration within the two days of competition we made a series of early decisions about the limitations of the OAI-ORE export and import plug-ins and a minimal acceptance test – the complete lossless transport of a full set of records from one repository software to another – was specified. The limitations concerning the rest of the implementation are listed below:

- RDF/XML resource map serialisation only.  
This not only speeds up the process of producing the resource maps that are exported, but more importantly it reduces the overhead of parsing the document when it is imported. RDF has a simple parser which was already available in both softwares.
- Dublin Core Metadata is source of primary metadata.  
Again, to reduce overhead and the amount of implementation needed we agreed that the Dublin Core metadata (conforming to the OAI\_DC specification [12]) would be sufficient when indexing the records sourced from another system. It was also assumed that both repository softwares had interpreted the meaning and definition of the Dublin Core terms in the same way. **Note:** This does not imply that non-Dublin Core metadata was lost, simply that it was not indexed in the destination repository at this time. This could easily be achieved later by implementing or enabling a series of importers which can understand other types of metadata representations, such as Fedora Object XML (FOXML) and EPrints XML.

### The EPrints export plug-in

Files and documents associated with each record/publication are listed using the built-in EPrints model which allows us access to each "eprint" object and a list of the files which are part of that "eprint." These files form the first part of the aggregation. To export the complete publication object we also need the history and provenance data which is revealed by the EPrints export plug-in manager. The EPrints export plug-in manager reveals all resources that can be exported through the EPrints interface including the history, provenance and metadata to be included in the OAI-ORE resource map. Listing all these resources along with the files in the aggregation guarantees the complete transport of each publication.

Finally the aggregation is encapsulated in an ad-hoc resource map which is generated on demand at a consistent URI. As the resource map of each EPrint is generated by the export manager, the resource map may contain a link to itself. It is necessary to handle this possible recursion on import.

### The EPrints import plug-in

The EPrints OAI-ORE import plug-in supports importing either a single XML resource map or a list of resource map locations (URLs). The imposed limitation was that this resource map had to contain a single aggregation (as per the OAI-ORE specification) representing a publication. In turn this aggregation had to contain a Dublin Core record conforming to the OAI\_DC specification. To import a list of resource maps the plug-in iterates through the input list of URLs, retrieving each one and then attempting to import it.

Each resource map is parsed for a component resource that /conformsTo/ the Dublin Core (DC) metadata schema (see "Extensions to the OAI-ORE Specification" section for information about the dc:conformsTo predicate). The built-in EPrints DC import plug-in is used to convert the provided DC record to an internal EPrints record. Each aggregated resource in the resource map is then retrieved and added to the EPrints record as a document. This includes the DC record XML as well as any other metadata resources provided such as FOXML.

At this stage in development in the EPrints import plug-in, recursive importing was not supported. This would be desirable in future versions such that you could hand the plug-in a top level resource map which contains an aggregation of all the resources in the repository. At this point it would be necessary to determine the type of resource before importing it, as they might not all be publications and each type might require a different import workflow as a result.

The EPrints import and export plug-ins that were used in this challenge on an EPrints 3.0 platform are available online [28]. It is intended that both of these plug-ins will come packaged with future versions of the EPrints software.

### The Fedora export plug-in

Due to the architectural design of Fedora, it was easier to create an independent application to produce OAI-ORE resource maps of the objects held in the repository than to attempt to extend Fedora's code with an export plug-in. Fedora's basic object model fits the OAI-ORE notion of aggregation, so that a one-to-one mapping between aggregations and Fedora objects is a semantically sensible assumption.

The application code to create resource maps was added to the Python-based Web interface powering the Oxford Research Archive repository [24]. The Python library RDFLib [26] provides the means to construct, hold and serialise RDF graphs, such as an OAI-ORE resource map. The service works by adding triples to an empty graph according to the OAI-ORE specifications, directly mapping the predicate ore:aggregates to each Fedora object's items, or 'datastreams' [7]. In Fedora, a datastream refers to any object which can be represented by a file, including provenance and history data.

Additional triples were added to the graph to further refine the information held in the resource map for each datastream, such as indicating the MIME type of the attachment and what standard the datastream might conform to (adopting the dcterms:hasFormat property for this purpose, but initially using it in a technically incorrect manner).

A built-in method of RDFLib was then used to serialise the resource map into whichever syntax was required, RDF/XML, ntriples [9], N3 [2], or Turtle [1] formats. The Atom format serialisation was added after the Open Repositories 2008 conference [21].

### The Fedora import plug-in

To enable importing OAI-ORE resource maps into Fedora, a script was used which first gathered the IDs of the items in the Open Repositories 08 EPrints repository and then ran through the list, maintaining a log of progress. The RDF/XML resource map for each EPrint object was downloaded and parsed by RDFLib. The resulting graph was then queried to find out the URLs of all the attachments and metadata files, and to find out which of these conformed most closely to simple Dublin Core. A Fedora object was instantiated for each EPrint object, and a process downloaded the associated files from EPrints and added them as datastreams to the surrogate Fedora object. The metadata file that conformed to simple Dublin Core as mentioned above was placed in a datastream with the ID of 'DC', which is the convention for Fedora objects.

### Extensions to the OAI-ORE Specification

As has already been hinted above, we found it necessary early on to include a small amount of Dublin Core relating to the aggregated resources themselves within each aggregation. With each publication aggregation containing many metadata records of different types, it is impossible to determine which object conforms to which metadata specification without parsing them individually. This leads to a solution that is computationally more expensive than it needs to be. The key to solving this problem is to extend the OAI-ORE model by adding the Dublin Core namespace to the aggregation (it already exists in the resource map), such that the dc:conformsTo and dc:hasFormat predicates can be used to state explicitly which namespace or specification and document format each object in the aggregation corresponds to. This allows the import plug-ins to quickly determine which object in the aggregation represents DC metadata.

Figure 6 shows an abstracted version of a resource map for a single publication as used in the demonstration. This example resource is taken directly from the Open Repositories Conference 2008 repository used as one of the demonstration repositories.

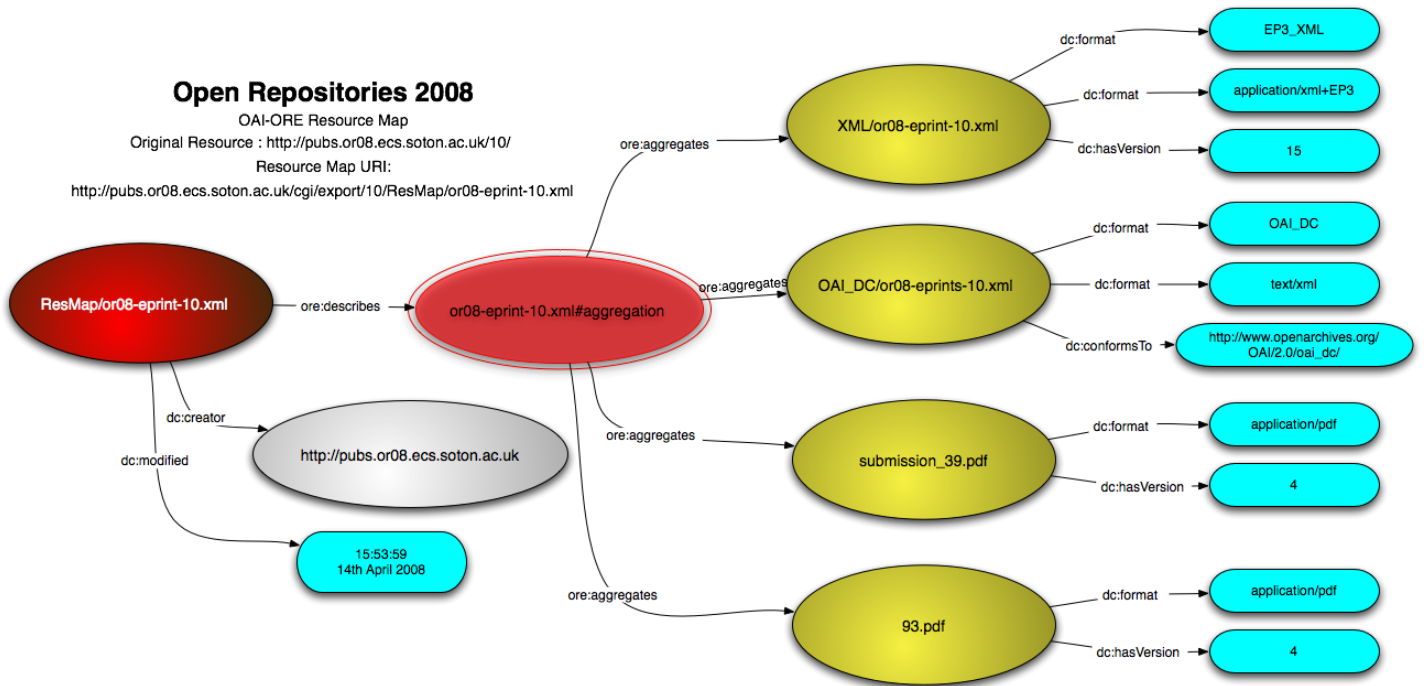


Figure 6: Complete OAI-ORE example Resource Map and Aggregation with Dublin Core extensions [View full-size image]

### EPrints -> Fedora

For this demo the live publications archive from the OR08 conference (in EPrints) was used as the source archive. This was then exported into the Fedora repository software in around 20 minutes. Figure 7 shows screenshots of the source and destination repositories in each software and displays the subject tree. This was recognised by the Fedora software with only minor changes to the software (telling Fedora which field to group and index on). As can be seen, not only is the content transported, but also the functionality and usability of the objects.



Figure 7: Screenshots of the OR08 repository transport, EPrints to Fedora. [View full-size image]

### Fedora -> EPrints

For this demonstration, the live publications archive from the Oxford Research Archive (in Fedora) was used as the source. Due to the size of the ORA it was decided to stop the import after about 35 minutes, which still resulted in the harvesting of a substantial number of records. Again screenshots of both source and destination repositories are shown. Figure 8 shows the same publication displayed by each repository. We can see a much more compressed citation in the EPrints repository and a greater number of files. These files represent the internal data from the source repository which only Fedora understands and can interpret; the same issue exists when moving content from EPrints to Fedora. To solve this problem a further series of import plug-ins could be developed to interpret this data in a fashion suited to the particular repository software.

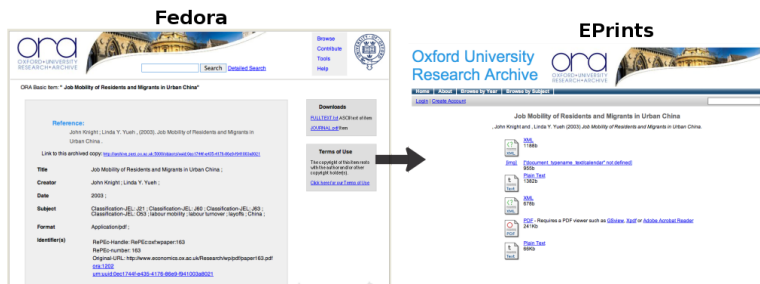


Figure 8: Screenshots of the ORA repository transport, Fedora to EPrints. More screenshots can be found here. [View full-size image]

It should be noted that the branding of each destination repository was done by hand and is not part of the OAI-ORE solution. This part of the demonstration was intended to show that a repository need not be stuck forever with its first choice of software. Rebranding is likely to be a key part of migrating to new repository software.

### OAI-ORE futures

Following the demonstration in April 2008, the final OAI-ORE specification [19] was released in November, and we hope that tools that use OAI-ORE to bind objects start to emerge in the near future. With such tools, OAI-ORE import support for the repository will become a mechanism for obtaining new resources, not just a tool to enable interoperability.

This paper has shown how OAI-ORE can be used to express resources at a relatively high level within the repository such that they can be transported and re-used elsewhere. This might imply that OAI-ORE is simply a high-level tool that is used on demand and then discarded. We feel, however, that the OAI-ORE model can be applied at a much lower level, becoming the basis on which repository software operates.

Current computing practices relating to low-level storage consist of a set of files on disk which, in the majority of cases, are bound together using an old office paradigm known as the folder. Files can only reside in a single folder unless they are copied. In other words, current filing systems contain no way to create custom collections of files without replicating the files themselves. OAI-ORE provides opportunities to bind low-level objects into multiple collections that can be used by higher level software while avoiding the need to make copies. The same object can be used in many aggregations that are related to many resource maps.

OAI-ORE is likely to gain momentum as different storage technologies, such as cloud storage and open storage [20], are used more widely. These platforms provide a "Drop Box" type technology where objects are assigned a UID or URI at the point they are stored. The objects in these platforms are all classed as individual resources and no low-level binding exists. Using UIDs and URIs for the object identifiers fits perfectly with the OAI-ORE model, where the URIs are required to apply OAI-ORE.

If OAI-ORE finds use in binding objects below any repository software, then we can envision how this demonstration would be different. Instead of using OAI-ORE to export and re-import the objects, we could simply change the software which sits on top of the objects. A little more work may also open the possibility of two repository softwares simultaneously managing the same set of low-level objects. At this point the definition of a repository as a single software solution starts to become abstracted into many layers involving basic storage and a set of services. This will also help from the point of view of repository services software being able to handle future technologies such as open storage with OAI-ORE bindings.

### Conclusion

The introduction of the OAI Protocol for Metadata Harvesting (OAI-PMH) was the first step in enabling repository interoperability, but it was not a complete solution. The OAI-ORE protocol adds new functionality while it is a completely separate standard that 'neither extends nor replaces' OAI-PMH. OAI-PMH has as its basic tenet "a mechanism for harvesting records containing metadata" from one repository for reuse elsewhere [13]. This has so far been adequate and supported the openness of repositories. However, as the uses of repositories and the types of content they contain expand, more comprehensive methods for sharing content and more capability in terms of what is harvested and how it is reused are required. This is where OAI-ORE comes into the picture.

This paper introduced the basic concepts of OAI-ORE and outlined how it can be used to express a single publication object or to build entire collections of objects. We showed how such representations can be serialised in both RDF/XML and the Atom format.

Implementing export plug-ins to reveal publication records as OAI-ORE Resource Maps in both the EPrints and Fedora platforms enabled us to write the corresponding import plug-ins. This then allowed us to demonstrate the complete transportation of the content from one repository into the other.

OAI-ORE provides another tool to help repository managers tackle the problem of long term preservation, providing a simple model and protocol for expressing objects so they can be exchanged and re-used. In future we hope to see OAI-ORE being used at the lowest level within a repository, the storage layer. Binding objects in this manner would allow the construction of a layered repository where the core is the storage and binding and all other software and services sit on top of this layer. In this scenario, if a repository wanted to change its software, instead of using OAI-ORE to migrate the objects from one software to another, we could simply swap the software.

## References

1. Beckett, D. (2004). Turtle – Terse RDF Triple Language. <http://www.dajobe.org/2004/01/turtle/>
2. Berners-Lee, T. (1998). Notation 3 – A readable language for data on the Web. <http://www.w3.org/DesignIssues/Notation3>
3. Berners-Lee, T. (2006). Linked Data – Design Issues. (Widely known as "The 4 Rules of the Web") <http://www.w3.org/DesignIssues/LinkedData.html>
4. Carr, L., White, W., Miles, S. and Mortimer, B. (2008) Institutional Repository Checklist for Serving Institutional Management. In: Third International Conference on Open Repositories 2008, 1-4 April 2008, Southampton, United Kingdom. <http://pubs.or08.ecs.soton.ac.uk/138/>
5. Common Repository Interface Group (CRIG), set up to examine the boundaries between repositories and other systems. <http://www.ukoln.ac.uk/repositories/digirep/index/CRIG>
6. Dublin Core Metadata Terms. <http://dublincore.org/documents/dcmi-terms/index.shtml>
7. Fedora Tutorial #1. Introduction to Fedora. <http://www.fedora.info/documentation/2.2.3/userdocs/tutorials/tutorial1.pdf>
8. Fielding, R. et al. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, Section 10, Status Code Definitions. <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
9. Grant, J. and Beckett, D. (2004). RDF Test Cases – ntriples. W3C Recommendation. <http://www.w3.org/TR/rdftestcases/#ntriples>
10. Gregorio, J., de hOra, B. (2007). The Atom Publishing Protocol. IETF RFC 5023. <http://tools.ietf.org/html/rfc5023>
11. Jacobs, I. and Walsh, N., Architecture of the World Wide Web (Volume One), W3C Recommendation. <http://www.w3.org/TR/webarch/>
12. Johnston, P., et al (2002). Open Archives Initiative Dublin Core (OAI DC), XML Schema. [http://www.openarchives.org/OAI/2.0/oai\\_dc.xsd](http://www.openarchives.org/OAI/2.0/oai_dc.xsd)
13. Johnston, P., et al (2008). Open Archives Initiative object re-use and Exchange. *ORE User Guide, Primer*. 11 July 2008. <http://www.openarchives.org/ore/primer>
14. Joint Information Systems Committee – Supporting education and research. <http://www.jisc.ac.uk>
15. Lagoze, C., Van der Sompeel, H., Nelson, M.L., et al. (2008). Object re-use and exchange: a resource-centric approach. <http://arxiv.org/ftp/arxiv/papers/0804/0804.2273.pdf>
16. Manola, F. and Miller, E., et al. (2004). RDF Primer. W3C Recommendation. <http://www.w3.org/TR/rdft-primer/>
17. Mining for ORE, Dave Tarrant, Ben O.Steen and Tim Brody. Video: <http://blip.tv/file/866653>
18. Open Archives Initiative Announces U.K. Public Meeting on April 4, 2008 for European Release of Object Reuse and Exchange Specifications. <http://www.openarchives.org/ore/documents/EUKickoffPressrelease.pdf>  
Open repositories 2008: OAI-ORE European Rollout Meeting. <http://or08.ecs.soton.ac.uk/ore.html>
19. Open Archives Initiative object re-use and Exchange, (2008). ORE Specification and User Guide. <http://www.openarchives.org/ore/loc>
20. Open Archives Initiative Protocol – Object Exchange and Reuse. <http://www.openarchives.org/ore/>
21. Open Repositories 2008 (OR08). <http://or08.ecs.soton.ac.uk/>
22. Open Repositories 2008 Digital Repository. <http://pubs.or08.ecs.soton.ac.uk/>
23. Open Repositories 2008: Developer Activities – The Repository Challenge. <http://or08.ecs.soton.ac.uk/developers.html>
24. Oxford University Research Archive (ORA). <http://ora.ouls.ox.ac.uk/>
25. Preserv: Inspiration and Provision of Preservation Services for Digital Repositories. <http://www.preserv.org.uk/>
26. RDFLib. A Python library for working with RDF. <http://rdflib.net/>
27. RepoChallenge. For details and videos of entries see: [http://www.ukoln.ac.uk/repositories/digirep/index/CRIG\\_Repository\\_Challenge\\_at\\_OR08](http://www.ukoln.ac.uk/repositories/digirep/index/CRIG_Repository_Challenge_at_OR08)
28. Tarrant, D., Brody, T. (2008) EPrints ORE Resource Map Import and Export Plug-ins. <http://files.eprints.org/353/>
29. Tarrant, D., O'Steen, B., Hitchcock, S., Jefferies, N. and Carr, L. (2008) Applying Open Storage to Institutional Repositories. In: 2nd European Workshop on the Use of Digital Object Repository Systems in Digital Libraries (DORS2L2), September 18th 2008, Aarhus, Denmark. <http://eprints.ecs.soton.ac.uk/16679/>

## About the Authors

David Tarrant is a researcher in the School of Electronics and Computer Science at the University of Southampton. His main areas of research are web, web science and their relations to digital scholarly material. Recent projects which he has worked on include Preserv2, which focuses on digital preservation and EPrints, the digital repository platform, of which he is a developer. Email: [dct05r@ecs.soton.ac.uk](mailto:dct05r@ecs.soton.ac.uk)

Ben O'Steen is an Institutional Repository Software Engineer at Oxford University Library Services. Email: [benjamin.osteen@ouls.ox.ac.uk](mailto:benjamin.osteen@ouls.ox.ac.uk)

Tim Brody is at the University of Southampton, where he develops the following repository services: Celestial (OAI harvester), Citebase (citation-ranked search service covering arXiv and other subject repositories), the Registry of Open Access Repositories (ROAR), and PRONOM-ROAR, a joint initiative with the UK National Archives and the JISC Preserv project to add a format identification service to ROAR. Tim was awarded a PhD at Southampton in 2006. Email: [tdb01r@ecs.soton.ac.uk](mailto:tdb01r@ecs.soton.ac.uk)

Steve Hitchcock is project manager for the JISC Preserv 2 project. He has a PhD (2002) from Southampton University, where he now works in the areas of digital publishing, digital preservation, open access and repositories. Email: [sh94r@ecs.soton.ac.uk](mailto:sh94r@ecs.soton.ac.uk)

Neil Jefferies is R&D Manager at Oxford University Library Services. Email: [neil.jefferies@ouls.ox.ac.uk](mailto:neil.jefferies@ouls.ox.ac.uk)

Les Carr is a senior lecturer at the University of Southampton, with expertise in Web-based information technologies. Les is a fellow of the Web Science Research Initiative, course leader for a new MSc in Web Technologies, and technical director of the EPrints repository software team. Email: [lac@ecs.soton.ac.uk](mailto:lac@ecs.soton.ac.uk)

Tags: OAI-ORE, repositories, standards

Subscribe to comments: [For this article](#) | [For all articles](#)

### One Response to "Using OAI-ORE to Transform Digital Repositories into Interoperable Storage and Services Applications"

Please leave a response below:

1. [felpeduca](#), 2009-09-22

Enhancement of repository services has been furthered by the adoption of RDF together with the OAI-ORE specification and its manifestation by the PRESERV2 team's activities that won them success in the OR08 Developer's Challenge. In the ORE world, the focus moves away from metadata and on to content, especially when dealing with the complexities of digital objects, resulting in better services for users. In addition to its other benefits, implementing OAI-ORE provides one tool in the repository managers' toolkit to help tackle the monumental problem of digital preservation, curation and continued access. The PRESERV2 model of preservation services provides one option for preservation activities to be undertaken in a way that is realistic and removes the prospect of a single gargantuan task to be undertaken at source. Coupled with OAI-ORE, such services can be applied to repository content of mixed complexity where details such as provenance and relationships are retained. Being able to invoke automated actions on digital objects in a way that is manageable and scaleable, and by enhancing object mobility so that more copies of those digital items can be stored in distributed locations, offers the increased likelihood that the objects will not only survive impresoras fiscales, but will remain accessible and by extension, usable. Once these preservation services and actions are realised, the focus then moves to policy.

