The ARROW project after 2 years: are we hitting our targets?

Andrew Treloar
Director, Information Management and Strategic Planning
Monash University
andrew.treloar@its.monash.edu.au

Geoff Payne[i]
Director Library Corporate & Financial Services
Latrobe University
g.payne@latrobe.edu.au

***Abstract**:*

This paper focuses on the institutional repository work of the ARROW project, which has been developed within the following three stage framework:

1. *conceptualising and developing an ARROW institutional repository solution comprising software, policy frameworks and implementation strategies,*
2. *implementing the ARROW repository within the project's partner institutions, and*
3. *offering the ARROW repository solution to other Australian universities.*

The paper will look at the decisions the ARROW project made at the outset and provide a review of those decisions after two years of operation. It will also look forward at ARROW's plans for 2006 and ongoing development work.

1. Introduction

The ARROW project commenced work in February 2004 with the objective of developing best of breed solutions for open access institutional repositories and electronic publishing in Australian universities over three years. The ARROW consortium consists of Monash University as the lead institution, Swinburne University of Technology, the University of New South Wales and the National Library of Australia. The ARROW institutional repository design goal has been to produce a generic storage mechanism for a combination of any type of digital content that might arise from the research efforts of Australian universities.

2. ARROW Technology Decisions[ii]

2.1. Original vision

The technology vision for ARROW was presented in the original bid document as a layered architecture diagram. Such layered architectures have been preferred since at least the days of the International Standards Organisation Open Systems Interconnect seven-layer reference model for network services (ISO 2005). In the digital library field these sorts of high-level models are so common, and appear so frequently, that the project group took to referring to 'obligatory' layered architecture diagrams. Figure 1 therefore is the OLAD (Obligatory Layered Architecture Diagram) for ARROW.

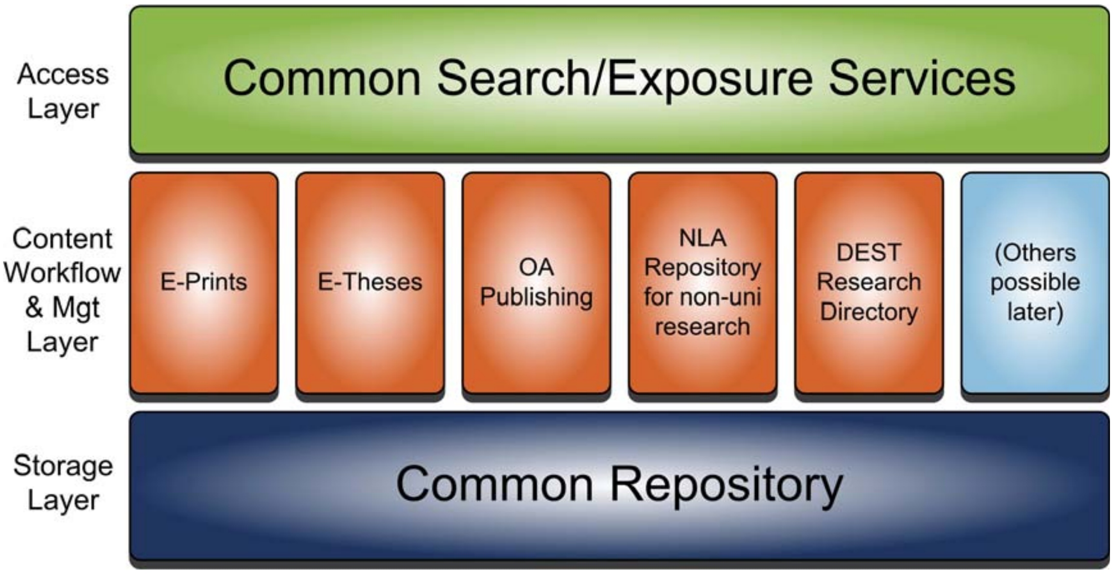

Figure 1: ARROW OLAD

So, what decisions did the project make to turn this high-level diagram into a working set of software?

2.2. Storage layer

The project recognised very early on that the choice of repository technology would determine the functionality ARROW could provide and the ways it could provide it. Much of the latter half of 2003 was spent in careful analysis of available candidates, looking in particular at the list and functionality checklist contained in the report from the Open Society Institute (Open Society Institute, 2004).

As a result of this work, the project decided to select Fedora™ – the Flexible Extensible Digital Object and Repository Architecture. This architecture came out of Digital Library work done in the computer science field in the late 1990s (Payette and Staples 2002). The current software implementation of this architecture is "an open source, digital object repository system using public APIs exposed as web services" (Staples, Wayland and Payette 2003).

Fedora™ can best be thought of as services-mediation infrastructure, rather than an off-the-shelf application. It can use web services technology (http://www.w3.org/2002/ws/) to draw on services provided by other systems as well as expose its own functionality using web services standards. Key to the Fedora™ architecture is its underlying object-based model. Fedora™ stores digital content objects, either as datastreams contained within the repository or as links to external resources. It also stores what Fedora™ calls disseminators. These are stored programs which provide ways to render these digital content objects. As an example, a Fedora™ repository might contain an image disseminator which can take a stored image object and render it on the fly into a thumbnail, a medium-resolution version or a high-resolution version as required. The software maintains bindings between content objects and their disseminators. Each object has a default disseminator (which might just provide the sequence of bits that comprise the object plus a Multi-purpose Internet Mail Extensions (MIME) type (http://www.ietf.org/rfc/rfc2045.txt), much like a web server). Alternatively, the repository might provide alternative disseminators that will allow the object to be exposed in other ways. An example of this might be a disseminator which exposes the internal structure of an Encoded Archival Description (EAD) (http://www.loc.gov/ead/) as a navigation mechanism. This architecture, which combines objects and disseminators, is very flexible, and provides significant advantages as a platform on which to build other applications (Lagoze, et. al. 2005). For more background on the reasons for selecting Fedora for the ARROW project, see Treloar (2005).

Fedora™ has released a number of versions of its software since the ARROW project started working with it:

- Version 1.2, released in late December 2003, provided versioning of both objects and their disseminators, as well as a Java-based Administration GUI.

- Version 2.0, released in late January 2005, added a new internal storage format called Fedora Object eXtensible Markup Language (FOXML), an Resource Description Framework (RDF) based resource index, significant performance tuning, relationship metadata for digital objects and a host of other minor changes.

- Version 2.1b, released in mid November 2005 (and expected to lose the beta-designation in December 2005), added a new security architecture, a revised service framework to make it easier to integrate software into Fedora, a new OAI provider service and a directory ingest service for easier bulk-loads. For further details, consult the release notes available online at http://www.fedora.info/download/2.1b/userdocs/distribution/release-notes.html.

Two years in, the decision to select Fedora™ has (despite some delays in successive versions of Fedora) proven to be a very sound one. The software is proving to be both flexible and extensible, as well as easy to build on top of. The project has not yet found anything that it wanted to implement that the software prevented on the basis of inherent design constraints. Given the Mellon funding for Fedora™ through to the end of 2007, and the growing user community in the US and Europe, the future of Fedora™ looks secure. In addition, VTLS (see below) have offered to maintain the Fedora™ code if this becomes necessary in the future.

2.3. Content workflow and management layer

Fedora™ is now a mature repository engine, with deep and rich functionality applicable to a range of problem domains. But it does not provide an out-of-the-box experience. When one downloads Fedora, one gets the engine only. To do something useful with Fedora, one has to write software. This was the intention of the Content Workflow and Management Layer.

ARROW decided early in 2003 to have the main modules written by VTLS (http://www.vtls.com). At the time, they had just announced a product, VITAL (http://www.vtls.com/Products/vital.shtml), built on top of Fedora. ARROW licensed VITAL and has been working with VTLS to extend the functionality of Fedora™ by commissioning a series of Open-Source Web Services. The current software for managing ingest into the repository is a Windows application called the VITAL Client Manager. Web-based content ingest was provided in VITAL 2.0 in late 2005, and web-based management should be available in 2006 as part of the releases of VITAL 3.0 and 4.0.

This decision to partner with VTLS had a number of advantages:

- It saved the ARROW project 3-6 months of startup time (hiring programmers and getting them up to speed on the Fedora™ APIs)
- It outsourced the much of the software development risk
- It provided a support base for the software beyond the life of the project
- It contributed to the functionality of the Fedora™ code base
- It ensured that the ARROW functionality benefits the global institutional repository community

In addition to the VITAL software, ARROW is also trialling the Open Journal Systems software (http://www.pkp.ubc.ca) from the Public Knowledge Project at Simon Fraser University and the University of British Columbia. The ARROW

OLAD shows the OA Publishing module sitting on top of a common repository. This was the original thinking behind the bid, but at present, OJS comes with its own filesystem-based repository. The long term plan is to integrate the workflow and exposure tools of OJS with the underlying Fedora™ repository engine. Ron Jantz at the Rutgers University Library Scholarly Communication Center (http://www.scc.rutgers.edu/scchome/) has already developed software to migrate content from OJS into Fedora™ for archival storage. ARROW is working to adapt the Rutgers software to work with the now updated versions of OJS and Fedora, and to share these enhancements with Rutgers.

The decision to partner with VTLS has proved to be a good one. This has been a true partnership, with significant transfers of intellectual property in both directions. ARROW has been able to influence the direction of future versions of VITAL, as well as getting access to pre-beta code for testing and critiquing. As well, at the end of the ARROW project, those universities which have licensed the VITAL software will have a ready-made sustainability solution, with VTLS providing ongoing support and upgrades.

The ARROW project deliberately commenced this software development path with a high level functional requirements document which it knew would need to be fleshed out as the project progressed. Recognising that the project was entering uncharted territory the core project team decided to work with a costing model based around an agreed budget of development points rather than a tightly specified development plan. This approach has served ARROW well as it has been able to reprioritise software developments by assigning development points from one area to another. This has given the project the flexibility to avoid duplicating functionality that was becoming available in Fedora™ in parallel with project development work.

2.4. Open source

It was a condition of the DEST funding for ARROW that any commissioned software would need to be made available as open source. ARROW has partnered with VTLS to commission them to write a series of ARROW-funded modules. These ARROW-commissioned modules will call Fedora™ using the existing Fedora™ web-services APIs and will also expose themselves as a series of web-services. VTLS will be able to build products on top of these new ARROW-commissioned modules if they wish and future releases of the VITAL product will almost certainly use these modules.

The announcement of the initial series of open-source modules was made in late September 2005, and the modules are now available for download from http://www.vtls.com/Products/osc.shtml. The modules include the following functionality:

- Automated capture of technical metadata for preservation purposes.
- Web Crawler Indexing and Exposure.
- Metadata extraction & automatic validation of content via JHOVE (JSTOR/Harvard Object Validation Environment - http://hul.harvard.edu/jhove/).

- Handles server integration for persistent identifiers.
- SRU/SRW (see below) Interface for exposure of repository content.

Further open-source modules will be developed and made available over the life of the project.

2.5. Access layer

The whole point of putting content into ARROW is to expose it as widely as possible. The project decided to provide five ways to expose ARROW content.

Firstly, each repository would have its own web interface. This is based on the VITAL Access Portal, but will be themed according to a combination of ARROW and institution brand. This interface will only display the records contained in that repository.

Secondly, the underlying Fedora™ software natively supports the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). This means that any OAI-PMH compliant harvesting software (such as OAISTER - http://oaister.umdl.umich.edu/o/oaister/) would be able to retrieve metadata about ARROW repository contents.

Thirdly, the ARROW National Research Discovery Service has been written by the National Library of Australia as part of their contribution to the ARROW consortium and is hosted on their Teratext system. It uses OAI-PMH to harvest metadata from a number of different institutional research repositories at Australian universities. These repositories use a range of software (e-prints.org software, DSpace and Fedora) but all expose their metadata for harvesting. This service is now live and available either through a link from the ARROW website or directly at http://search.arrow.edu.au/. This service allows for searching research outputs across the Australian university sector.

A fourth exposure mechanism is having the content indexed by web search engines. The VITAL software exposes repository content in a way that facilitates this. The ARROW project has been working particularly with Google, but the approach taken is one that will work with a range of other web indexing systems. Part of the Google development has been to allow Google to index content that is not publicly available to improve its retrieval. This is initially going to be used for thesis content, where Monash does not want to make the electronic version of a thesis available for direct download, but does want to make it discoverable for request via inter-library loan.

The fifth exposure mechanism is to make it possible for other search services to connect directly to ARROW repositories and run interactive searches. The standard protocol for such connections in the library world is Z39.50 (More formally known as ISO 23950: "Information Retrieval (Z39.50): Application Service Definition and Protocol Specification") (http://lcweb.loc.gov/z3950/agency/). Z39.50 has not been taken up as quickly as its proponents had hoped (for a variety of reasons too complex to cover here). As a result the Z39.50 Next Generation group (ZNG) have been working on more modern and lightweight protocols to achieve much of the original Z39.50 functionality. These newer protocols are called SRU (Search/Retrieve over

URL) (http://www.loc.gov/z3950/agency/zing/srw/sru.html) and SRW (Search/Retrieve for Web Services) (http://www.loc.gov/z3950/agency/zing/). ARROW decided to support both SRU and SRW connections as they are well suited for searching otherwise complicated protocols or proprietary access mechanisms. Since the search results and other information are transferred using XML, SRW/SRU has become a tool for creating bridges to more technically complicated protocols, such as Z39.50.

The wide range of exposure choices has also worked well. Organisations have been able to use these in a variety of ways to access and expose ARROW content. One additional mechanism that may be considered in 2006 is the provision of targeted RSS feeds, but more content will be needed before this will be attractive to end users.

3. ARROW Implementation Decisions

As well as making technology decisions, the ARROW project needed to make a whole series of implementation decisions. Those that have required most attention are decisions around scope, use-cases and content modelling, descriptive metadata, and persistent identifiers.

3.1. Scope Constraints

At the outset the ARROW project decided to impose some constraints on its immediate aspirations as an essential part of confining the scope of the project to something achievable, and to allow delivery of sufficient functionality to get started on populating ARROW repositories as soon as possible.

The first constraint agreed was that ARROW would support digital object storage and retrieval, but not authoring. While using a standard authoring environment can improve the homogeneity of digital objects, it was felt that the advantage of this approach would become more evident once the project could demonstrate the ARROW repositories' search functionality.

The second self-imposed constraint was that the project would concentrate, at least in the first instance, on capturing digital objects that could be rendered by a web browser. One of the distinguishing features of Fedora™ is the capability to associate behaviours with digital objects. This is achieved by developing and including software in the repository referred to as a "disseminator" for manipulating a set of associated digital objects. This capability was seen as of secondary priority until the software could manage the well known content types that can be displayed by a web browser. As the project has progressed this constraint has been softened somewhat, and VITAL 2.0 provides capabilities for manipulating JPEG2000 and MrSID images, and a page turning interface for viewing Portable Document Format (PDF) documents. There has also been discussion about providing a specialised disseminator for viewing high resolution X-RAY images as a pre-condition before they could be used in the context of ARROW repositories.

The constraints imposed by the absence of established conventions for content models (see below) are manageable while the project is working with objects that can be rendered by a web browser. For example in ARROW repositories a PhD thesis can be stored as any number of datastreams (that is individual pieces of digital content) each of which can be rendered by a web browser. Thus the repository can store an abstract and the body of the text as individual chapter by chapter PDF files, or it can store the whole text as a single PDF file. ARROW can also store accompanying materials such as digital audio interview files, digital images and digital video clips. Each of these can be rendered one at a time by the users' web browser. However capturing a collection of documents from different repositories may well create problems if they have been modelled differently in each repository. For example if the abstract for each thesis was not encoded as a separate datastream then the aggregate set could not consistently display search results at the abstract level.

The most difficult constraint on ARROW repositories at present arises in relation to access control. Access control for a repository typically places restrictions on what individuals can locate via a search or retrieve, but may also encompass restrictions on what repository management functions they can invoke. In the VITAL 2.0 software access control is supported only as an on/off switch for individual datastreams, with no variation possible based on user attributes. Fedora™ 2.1 supports the eXtensible Access Control Mark-up Language for access control (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml). This allows the incorporation of rules for access to individual datastreams of digital objects based on policies around user attributes and metadata associated with each datastream. Unfortunately there is no standard vocabulary for expressing access control conditions within XACML, and this will be essential if federated searching of repositories is to be possible with predictable results. A further problem relates to the assumption of Shibboleth as the regime for sharing and assigning user attributes which can be presented as input to decision making about access for individuals. Shibboleth is not widely adopted amongst Australian universities as yet. Work to address these issues is beginning in conjunction with Fedora, VTLS, the Metadata Access Management System (MAMS) project and the JISC repositories program in the hope that these participants can collectively develop a consistent vocabulary for expressing access conditions in repositories.

3.2. Use cases and content modelling

One of the advantages of Fedora™ is the lack of constraints it imposes on data structures within a Fedora™ repository. The price one pays for this flexibility however, is that it is necessary to design a content model for each distinct class of digital objects one wishes to store. A content model specifies the number and type of datastreams that must be present in an object, the metadata datastreams that must be present, and how to validate each datastream as a particular file format or conformant with a particular metadata schema.

The design of a content model may be arbitrary, however it is best to consider the ways in which the objects will be manipulated as part of the design process. For example the level to which a digital object is separated into component pieces (its granularity) directly sets the level at which individual parts can be referenced or reused in other objects.

For searching purposes metadata needs to be included to support views of the repository. For example, metadata about an author's faculty is required if an institution wants to be able to present the research outputs generated by a given faculty.

The preparation of use cases is the process of establishing descriptions of the ways the repository will be used, the access conditions required on whole objects and individual parts of objects, and the metadata that needs to be stored with the objects to allow particular searches and access constraints to be supported.

Only after one has produced a set of use cases can one begin the process of designing how an object will be stored in Fedora™. This process is known as "content modelling".

In the VITAL 2.0 software ARROW has implemented content models for seven different object types: journal articles, conference papers, working papers, books, book chapters, theses and images. The datastreams and required metadata for these object types have been documented in a spreadsheet. Templates have been produced to capture required datastreams and metadata to ensure the consistent structure of each member object of any one of the seven types supported.

The content model must also specify the naming conventions for datastreams. For example, when an access policy needs to be enforced at the object level, the datastream in which metadata can be found to inform the required decision must be consistently named across all objects in the repository.

It is the project's aspiration to generalise the declaration of a content model in software, from which will follow the data capture template and the display and access policies, and requirements for batch ingest and so forth. This generalisation is required to allow for the variety of different digital objects that the project will be required to manage as an institutional repository beyond the initial seven object types named above. Undertaking this work is part of the ARROW work plan for 2006.

3.3. Descriptive metadata

Early in the project ARROW spent some months examining the idea that a single descriptive metadata schema for all the objects in the ARROW repositories would be a sensible goal. After looking at the strengths and weaknesses of numerous metadata schemas, and upon considering the diversity of object types ARROW repositories could be required to store, it was decided that it was more realistic to accept that the project would need to support multiple descriptive metadata schemas.

As a result, ARROW has decided to support the metadata generated by communities of practice to accompany their digital objects. To do otherwise would risk reducing the accessibility of the objects and increase the likelihood of their remaining outside the ARROW repositories as a consequence.

The ARROW Research Discovery Service (http://search.arrow.edu.au/), developed by the National Library of Australia, harvests Dublin Core metadata from Australian research repositories to provide federated search and alert services across their content. The decision to support multiple descriptive metadata schemas within ARROW requires mappings between these schemas and Dublin Core to populate the discovery service. The project has decided to achieve this via the OCLC metadata interoperable core technology (Godby et. al. 2003), and discussions have been proceeding with OCLC to this end for some time.

To take advantage of the “native” metadata associated with collections of objects in ARROW repositories, the 2006 work plan includes implementing SRW/SRU search capabilities to allow searching of native metadata for each collection of objects in ARROW. SRW/SRU supports an explain transaction which can discover what metadata fields are associated with an object collection, and this can then populate a search screen to capture users’ search criteria.

3.4. Persistent identifiers

After careful consideration of all the available alternatives, ARROW decided to use handles for all the partner university sites. The NLA decided to proceed using its existing persistent identifier scheme. The Handles System (http://www.handle.net/), developed by CNRI (http://cnri.reston.va.us/), is a comprehensive system for assigning, managing, and resolving persistent identifiers, known as handles, for digital objects and other resources on the Internet. Handles can be used as Uniform Resource Names (URNs). Part of the work done by VTLS and released as Open Source has been the addition of handles functionality to the Fedora software.

The ARROW repositories were designed from the beginning to be as flexible as possible. To this end, the project decided it would be good practice to be able to persistently cite both objects and components of objects. The ARROW software therefore assigns handles to each entire ARROW object (such as a thesis), and to each component of an ARROW object (such as the metadata, the thesis abstract, the thesis body, and the reference list). This means that repository managers can disaggregate and re-aggregate objects as required in the future without the user being aware of it. It also means that the minimum persistently citeable unit can be made as granular as is required.

4. Aspirations for 2006

Apart from the additional developments referred to above, the primary aspiration for 2006 is to achieve the embedding of the ARROW repositories into the information management practice of participating universities. This process will be enhanced by the implementation of ARROW repositories in a wide cross section of Australian universities, either by their joining the ARROW project, or by their participation in related projects such as the Regional Universities Building Research Infrastructure Collaboratively (RUBRIC) project. The ARROW team is looking forward to the fruits arising from bringing more minds to bear on the remaining challenges. As well as this embedding, there are two specific ways in which ARROW will be extended in 2006.

4.1. DART additions

In August 2005, DEST announced the projects that had been successful in the Managed Environment for Research Repository Infrastructure (MERRI) round of funding. One of the funded projects was the Dataset Acquisition, Accessibility and

Annotation e-Research Technologies (DART) project (http://dart.edu.au/). This project will undertake a co-ordinated programme of eResearch requirements analysis, software development, policy and guideline creation and prototyping to investigate how best to:

- collect, capture and retain large data sets and streams from a range of different sources;
- deal with the infrastructural issues of scale, sustainability and interoperability between repositories;
- support deposit into, access to, and annotation by a range of actors, to a set of digital libraries which include publications, datasets, simulations, software and dynamic knowledge representations;
- assist researchers in dealing with intellectual property issues during the research process;
- adopt next-generation methods for research publication, dissemination and access.

A number of the work packages in DART will build on the work already done by ARROW (for details of the work packages, please consult the public bid document on the DART web site).

Work packages SI1 and SI2 will extend Fedora™ by adding support for the Storage Resource Broker (SRB). SRB is data storage virtualisation middleware that supports very large datasets on a range of storage devices. SRB abstracts the details of the individual storage devices into a virtual storage layer.

Work package CR5 will extend the ARROW software offering to support deposit of large datasets into institutional repositories, either on their own, or linked to publications.

Work package DA1 will build on the work being done by ARROW on access control to allow users to control access to deposited datasets.

Work package DA2 will improve repository deposit rates, sharing and re-use by improving discoverability of datasets. It will do this by extending the ARROW National Research Discovery Service to also harvest metadata about datasets as well as publications.

Work packages AA1, AA2 and AA3 will allow researchers to annotate each other's works (including publications and datasets stored in ARROW repositories), improve rates of annotation by allowing end-user control over who can annotate what and who can access the annotations, and work to assist collaborative annotation services to contribute to the life and productivity of research communities.

4.2. RQF support

The Research Quality Framework for Australian Universities (RQF) (http://www.dest.gov.au/sectors/research_sector/policies_issues_reviews/key_issues/research_quality_framework/) is an initiative of the Commonwealth Government to improve the assessment of the quality and impact of publicly funded research.The impending introduction of the RQF strengthens the case for institutions to capture their research outputs into repositories. Once captured they can readily be accessed by university staff who can select the work to be submitted for evaluation under the RQF, as well as be accessed by the RQF assessment panel members.

Monash University is planning to use the DART annotation facility in conjunction with the existing ARROW infrastructure to support a Mock-Research Quality Framework (RQF) process in 2006 and perhaps the actual RQF in 2007. For the 2006 Mock-RQF, Monash is planning to use ARROW to gather and store the electronic versions of the research outputs, implement access-controls to expose the selected outputs to the assessment panels, and use the DART annotation functionality to allow the assessment panels to make their comments. This system will be fine-tuned as the Government provides further advice about how the RQF will work.

4.3. Rollout

The ARROW roll out formally commenced in November 2005 when potential early adopters beyond the original ARROW consortium partners gathered at Monash University to explore how they might adopt the ARROW repository solution in the near term. Roadshows to showcase ARROW are planned for February 2006 at which time further institutions are expected to adopt the ARROW solution. ARROW's utility as a tool in the RQF process is expected to attract much interest.

5. Conclusion

At the end of 2005, ARROW has developed a workable institutional repository solution which with the release of VITAL 2.0 meets many of the requirements for an institutional repository capable of storing a wide range of digital objects. Our development priorities have necessitated some functionality being scheduled for development in 2006. With the 2006 workplan including XACML access control implementation and a variety of other improvements, the project team are confident ARROW can meet the needs of many Australian universities for an institutional repository platform.

There are some constraints arising from the immaturity of practice around repositories, and the consequent absence of standards across the sector. This is particularly evident for content models and fine grained access management, and this does not bode well for federated searching of repositories. A joint APSR/ARROW workshop to be held at the University of Sydney in early 2006 should start to address some of these issues in a co-ordinated way.

References

Dempsey, Lorcan, Eric Childress, Carol Jean Godby, Thomas B. Hickey, Andrew Houghton, Diane Vizine-Goetz, and Jeff Young. 2005. "Metadata Switch: Thinking About Some Metadata Management and Knowledge Organization Issues in the Changing Research and Learning Landscape." In *E-Scholarship: A LITA Guide*, ed. Debra Shapiro. February 2005 preprint available at http://www.oclc.org/research/publications/archive/2004/dempsey-mslitaguide.PDF

Godby, Carol Jean, Devon Smith, and Eric Childress. 2003. "Two Paths to Interoperable Metadata." Paper presented at the 2003 Dublin Core Conference, DC-2003: Supporting Communities of Discourse and Practice—Metadata Research & Applications, September 28-October 2, in Seattle, Washington (USA). Available at http://www.oclc.org/research/publications/archive/2003/godby-dc2003.PDF

International Standards Organisation 2005. *Open Systems Interconnection Reference Model*. Standards available online at http://www.iso.org/iso/en/CatalogueListPage.CatalogueList?ICS1=35&ICS2=100&scopelist=

Lagoze, et al. 2005. Fedora: An Architecture for Complex Objects and their Relationships. *Journal of Digital Libraries Special Issue on Complex Objects*. In press. Preprint available at http://www.arxiv.org/abs/cs.DL/0501012

Payette, Sandra & Staples, Thornton 2002, "The Mellon Fedora™ Project: digital library architecture meets XML and web services", Sixth European Conference on Research and Advanced Technology for Digital Libraries. *Lecture notes in computer science*, vol. 2459. Springer-Verlag, Berlin Heidelberg New York 406-421. Available at http://www.fedora.info/documents/ecdl2002final.PDF

Payne, Geoff, 2005. “Australian Research Repositories Online To The World – ARROW”, Educause Australasia 2005, Auckland, 7 April 2005. Available at. http://arrow.edu.au/docs/

Staples, Thornton, Wayland, Ross & Payette, Sandra (2003), "The Fedora™ Project: an open-source digital object repository management system", in D-lib Magazine, April 2003. Available at http://dlib.org/dlib/april03/staples/04staples.htm

Treloar, Andrew. (2005). "ARROW Targets: Institutional Repositories, Open-Source, and Web Services". In *Proceedings of AusWeb05, the Eleventh Australian World Wide Web Conference*, Southern Cross University Press, Southern Cross University, July. Available from http://ausweb.scu.edu.au/aw05/papers/refereed/treloar/.

Acknowledgements

The ARROW project is funded by the Australian Commonwealth Department of Education, Science and Training (http://dest.gov.au/), under the Research Information Infrastructure Framework for Australian Higher Education.

The DART project has been funded by the Australian Commonwealth Department of Education, Science and Training (http://dest.gov.au/ through the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovation Action Plan for the Future (BAA) (http://backingaus.innovation.gov.au/).

The authors also wish to thank the anonymous reviewers whose comments and questions have improved the paper.

Notes

[i] Geoff Payne was the ARROW Project Manager from early 2003 to November 2005.

[ii] For further details on the technologies mentioned in this paper, please either consult the referenced publications or the ARROW glossary at http://www.arrow.edu.au/about/glossary