For the love of metadata? : a functional approach to describing scholarly works

Julie Allinson, Repositories Research Officer, UKOLN, University of Bath

Rationale

Repositories are springing up across institutions in the UK and worldwide. For institutional repositories there is a pressing need to fill them with content and to make those contents available through search interfaces, aggregators and other services. Speed and easy access are paramount both for depositors, who want to add their materials to the repository with minimum effort, and for researchers, who want to discover the quickest route to the full-text. Consistent, good quality metadata is needed to provide the signpost to full-texts, yet there is a resulting tension between the effort required to create, and share, metadata and the needs of depositors.

Research and scholarly outputs are one of the main content types collected and managed by institutional repositories, in particular the research papers, or scholarly works, produced by academics and researchers[1]. In May 2006, the Joint Information Systems Committee (JISC) engaged the Eduserv Foundation and UKOLN to produce an application profile for scholarly works [1] that would facilitate the sharing of richer metadata between repositories and aggregators such as the newly-funded Intute search project [2]. This article describes the development of the application profile.

Metadata is not created for pleasure; it is created to serve a purpose and in order to know what this purpose was, we begun by drawing up a comprehensive list of functional requirements [3], an essential step in any development process. Amongst these, the most important were:

- Providing richer, more consistent metadata.
- Facilitating search, browse or filtering by metadata elements such as journal title, peer-review status and resource type.
- Identifying the latest, or most appropriate, version and facilitating navigation between versions.
- Supporting added-value services, particularly those based on the use of OpenURL ContextObjects
- Implementing an unambiguous method of identifying the full text(s).
- Enabling identification of the research funder and project code.
- Facilitating identification of open access materials[2].

In current practice, repositories mainly expose simple Dublin Core [4] records over OAI-PMH (the Open Archives Initiative Protocol for Metadata Harvesting) [5]. Simple DC does not offer the richness necessary to fulfil these functional requirements – a new approach to metadata was required.

---

1 In the context of this work an eprint or a scholarly work is defined to be a *scientific or scholarly research text* (as defined by the Budapest Open Access Initiative - http://www.earlham.edu/~peters/fos/boaifaq.htm#literature), for example a peer-reviewed journal article, a preprint, a working paper, a thesis, a book chapter, a report, etc.

2 Note that by 'open access' we mean "*free availability on the public internet, permitting any users to read, download, copy, distribute, print, search, or link to the full texts of these articles, crawl them for indexing, pass them as data to software, or use them for any other lawful purpose, without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself.*" http://www.earlham.edu/~peters/fos/boaifaq.htm#openaccess

The application model

The next step in our approach was to develop an application model. This is an important step as it allows us to identify what we are trying to describe - the entities - and the relationships between these entities. For bibliographic records, FRBR [6] already provides a useful model and this proved to be a good basis for developing our scholarly works model. The application model [7] is shown in figure 1 and illustrates the five entities, and the key relationships between them.

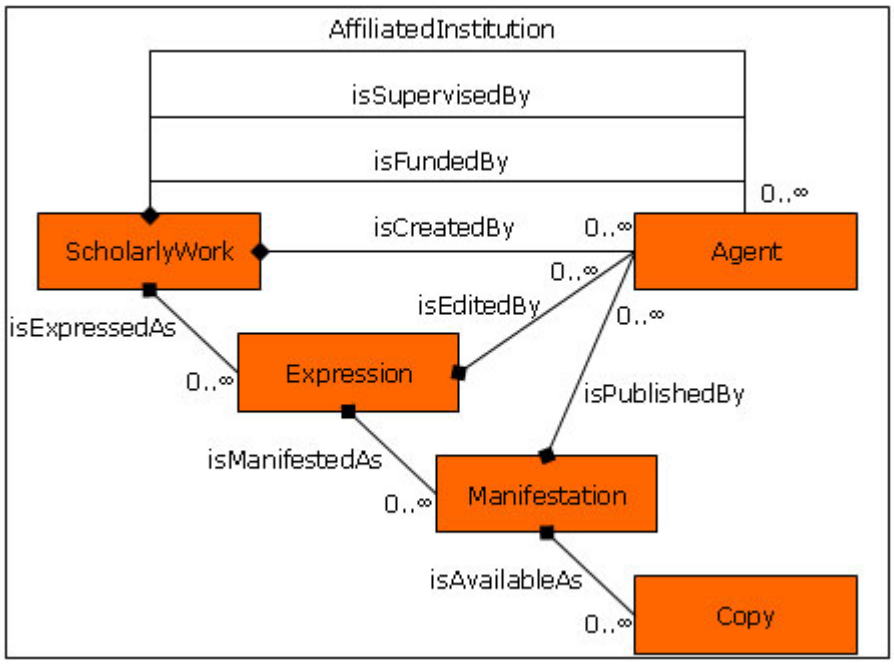

Figure 1 : the application model

According to our model, a ScholarlyWork is an abstract concept, effectively the intellectual 'idea' of the research paper. A ScholarlyWork may be expressed as one or more Expressions where Expressions are realisations of a ScholarlyWork, such as different revisions or translations. Each Expression may be manifested as one or more Manifestations, or formats, e.g. a Word document, a PDF, a HTML page. Finally, each Manifestation may be made available as one or more Copies, a html page in a particular network location for instance. The final entity, Agent, may be used to specify the creators, publishers, funders and other related persons or organisations engaged in the creation of the work.

In addition to these vertical relationships, the model also captures horizontal relationships (not shown on the above diagram), between related works, or between Expressions. These relationships enable the identification of connections between the various 'revisions' that a paper may go through (draft, pre-print, ..., final published version, etc.) and its different language translations.

To express this as metadata we define the key attributes required to describe each entity. For instance, the Expression entity includes attributes for bibliographic citation, genre / type, references, language and at the Copy level we capture licence and access rights. The application model documentation identifies the full list of entities, relationships and attributes. At this stage the model and its attributes are not tied to any particular metadata format and from the application model we move to the application profile, where metadata properties adhering to a particular metadata format are specified.

The application profile

The application profile [8] provides a way of describing the attributes and relationships of each of the five entities with metadata properties as part of a *description set*. Dublin Core properties are available to capture many of these. DC metadata is sometimes only considered capable of describing flat, single-entity, constructs like a single web page or document. However, the DCMI Abstract Model [9] introduces the notion of a *description* set, a group of related *descriptions*, which allows it to be used to capture metadata about more complex sets of entities, using application models like the one described above. DCMI is currently developing a revised set of encoding guidelines for XML and RDF/XML [10], which will allow these more complex, multi-description, *description set* constructs to be encoded and shared between software applications.

Description of the eprint as a ScholarlyWork		**Description of a Manifestation of an Expression of the eprint**
Entity Type (dc:type)		Entity Type (dc:type)
Title (dc:title)		Format (dc:format)
Subject (dc:subject)		Date Modified (dcterms:dateModified)
Abstract (dcterms:abstract)		Publisher (dc:publisher)
Identifier (dc:identifier)		Is Available As (eprint:isAvailableAs)
Creator (dc:creator)		**Description of a Copy of a Manifestation of an Expression of the eprint**
Funder (marcrel:FND)		Entity Type (dc:type)
Grant Number (eprint:grantNumber)		Access Rights (dcterms:accessRights)
Supervisor (marcrel:THS)		Licence (dcterms:licence)
Affiliated Institution (eprint:affiliatedInstitution)		Date Available (dcterms:dateAvailable)
Has Adaptation (eprint:hasAdaptation)		Is Part Of (dcterms:isPartOf)
Is Expressed As (eprint:isExpressedAs)		**Description of an Agent**
Description of an Expression of the eprint		Entity Type (dc:type)
Entity Type (dc:type)		Name (foaf:name)
Title (dc:title)		Family Name (foaf:family_name)
Description (dc:description)		Given Name (foaf:givenname)
Identifier (dc:identifier)		Workplace Homepage (foaf:workplaceHomepage)
Date Available (dcterms:dateAvailable)		Mailbox (foaf:mbox)
Status (eprint:status)		Homepage (foaf:homepage)
Version Number or String (eprint:version)		
Language (dc:language)		
Type (dc:type)		
Copyright Holder (eprint:copyrightHolder)		
Has Version (dcterms:hasVersion)		
Has Translation (eprint:hasTranslation)		

Bibliographic Citation (dcterms:bibliographicCitation)		
References (dcterms:references)		
Editor (marcrel:EDT)		
Is Manifested As (eprint:isManifestedAs)		

Figure 2 : the application profile metadata properties

The application profile lists the metadata properties, including mandatory elements, for each entity, provides usage guidelines and offers illustrative examples. Note that for this application profile, we have made very few elements mandatory. The profile makes use of properties from both the DC Metadata Element Set (simple DC) [3] and DC Metadata Terms (includes qualified DC terms) [11]. In addition, the MARC relator codes [12] and the Friends of a Friend (FOAF) Scheme [13] introduce additional properties; only five new properties have been created from scratch: grant number, affiliated institution, status, version and copyright holder. Figure 2 lists the metadata properties for each entity.

Where existing dc:relation qualifiers have been used (is part of, has version), the relationships being documented have been clearly defined alongside five new properties:

- has adaptation
- has translation
- is expressed as
- is manifested as
- is available as

Vocabulary Encoding Schemes facilitate the creation of consistent metadata and can help fulfil a number of the functional requirements. For this application profile, four vocabularies have been defined for:

- access rights (Open, Restricted or Closed)
- entity type (ScholarlyWork, Expression, Manifestation, Copy or Agent)
- status (Peer Reviewed or Non Peer Reviewed)
- resource type (see figure 3)

Figure 3: the eprints type vocabulary

The future

The application profile isn't a blueprint for repository design, rather it offers a mechanism for exchanging and exposing metadata records that can support the functional requirements outlined above. To facilitate this, an XML format known as Eprints DC XML [13] has been provided. This format is based very closely on the latest draft encoding guidelines for XML and RDF/XML being considered by the DCMI Architecture Community [14]. For existing repositories wishing to work with the profile, they may find that they already capture much of the richness specified here, but at the moment are unable to share this over OAI-PMH due to the limitations of simple Dublin Core. New repositories can use the application profile as a guide for making decisions about what metadata format to use, in conjunction with their local requirements. The application profile makes no statements about how a repository should create and store its metadata, nor about where information is drawn from. Increasingly repositories will be able to draw data from existing systems or use automatic generation tools, making the metadata process as efficient and authoritative as possible.

For this application profile to be useful, it must be validated, disseminated and discussed by the community. It also needs to be implemented in repositories and eprints dc xml made available for use by aggregators offering added-value services. Developers from the main repository software platforms, EPrints, DSpace and Fedora , have all indicated a willingness to work with the application profile and discussions within the community have indicated that there is interest and support for embedding the profile into the repository realm.

References

1. Eprints Application Profile http://www.ukoln.ac.uk/repositories/digirep/index/Eprints_Application_Profile
2. Intute Repository Search Project http://www.intute.ac.uk/projects.html
3. Eprints Application Profile Functional requirements specification

http://www.ukoln.ac.uk/repositories/digirep/index/Functional_Requirements
4.*Dublin Core Metadata Element Set*, Version 1.1. DCMI Recommendation. April 2006. http://dublincore.org/documents/dces/
5.Lagoze, Carl, Van de Sompel, Herbert, Nelson, Michael and Warner, Simeon. *The Open Archives Initiative Protocol for Metadata Harvesting*. Protocol Version 2.0 of 2002-06-14. http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm
6.IFLA, *Functional Requirements for Bibliographic Records*, 1998 http://www.ifla.org/VII/s13/frbr/frbr.pdf
7.Eprints Application Model http://www.ukoln.ac.uk/repositories/digirep/index/Model Powell, Andy, Nilsson, Mikael, Naeve, Ambjörn and Johnston, Pete, DCMI Abstract Model. DCMI Recommendation, May 2005 http://dublincore.org/documents/abstract-model/
8.Eprints Application Profile http://www.ukoln.ac.uk/repositories/digirep/index/EPrints_Application_Profile
9.Johnston, Pete and Powell, Andy. Expressing Dublin Core metadata using XML. DCMI Working Draft. May 2006. http://dublincore.org/documents/2006/05/29/dc-xml/
10.DCMI Usage Board, *DCMI Metadata Terms*, DCMI Recommendation, December 2006 http://dublincore.org/documents/dcmi-terms/
11.Library of Congress Network Development and MARC Standards Office, MARC Code Lists for Relators, Sources, Description Conventions, January 2007 http://www.loc.gov/marc/relators/
12.Brickley, Dan and Miller, Libby, *FOAF Vocabulary Specification*, January 2006 http://xmlns.com/foaf/0.1/
13.Johnston, Pete. *Eprints DC XML*. November 2006. http://www.ukoln.ac.uk/repositories/digirep/index/Eprints_DC_XML
14.DCMI Architecture Community. http://dublincore.org/groups/architecture/