

# A Registry of Collections and their Services: from Metadata to Implementation

Ann Apps

MIMAS, University of Manchester, M13 9PL, UK  
ann.apps@man.ac.uk

**Abstract:** The JISC Information Environment Service Registry (IESR) is a machine-to-machine middleware shared service providing a single central catalogue of quality descriptions of collections of resources available to researchers, learners and teachers in the UK, along with details of the services that provide access to those collections. The collections and services are described according to a set of metadata, which is defined by IESR, but is based on open standards wherever possible. The prototype registry is implemented as an XML repository indexed with the Cheshire II information retrieval software, with an associated meta-registry to support browsing and data capture. Several interfaces for server-to-server retrieval of IESR XML descriptions are available, as well as a Web interface.

**Keywords:** collection description, service metadata, machine-to-machine interface, Dublin Core, registry, meta-registry

## 1 Introduction

The JISC Information Environment Service Registry (IESR) [1] contains information about collections of resources available to researchers, learners and teachers within UK Higher and Further Education. Along with the collections are technical details of the services that provide access, as well as details of the parties that own the collections and administer the services.

Additionally IESR includes ‘transactional services’, which are not based on an explicit collection but provide a significant service, for example an institution’s OpenURL [2] resolver.

IESR is primarily a machine-to-machine middleware shared service within the JISC Information Environment [3]. It provides a single central catalogue of resources and their access details to portals and virtual learning services, removing the need for multiple copies of this information. A portal can discover a collection of interest to an end-user, possibly within a particular subject domain; determine the best access option; and provide to the end-user a link to the collection or a distributed search including it.

The data within IESR is supplied by collection and service administrators, thus assuring its quality. The IESR content manager makes a further quality check on supplied data. In addition to its machine-to machine interfaces, IESR has a Web interface to assist in content checking.

## 2 IESR Entities and Metadata

The design of the metadata used within IESR to describe resources is based on the Research Support Libraries Programme (RSLP) Collection Description schema (RSLPCD) [4]. This was developed to describe both physical and electronic collections within a wide range of domains,

including museums, archives and libraries. IESR describes electronic collections for the primary purpose of discovery. Thus the IESR data model is a simplification of the RSLPCD model, omitting details that seemed extraneous.

The IESR data model comprises three types of entity: a collection; a service, either informational (i.e. providing access to a collection), or transactional; and an agent that is the owner of a collection or an administrator of a service. A collection may have many services that provide access but it must have at least one service registered in IESR. An agent may be an owner or an administrator, or both, of many collections or services. It should be noted that within IESR the term 'service' is used to denote a single, low-level, technical access point to a collection. It is actually a conflation, made for pragmatic reasons, of the 'location' of the collection and a 'service' provided at that location. The underlying data model is described in more detail in [5].

The metadata properties used to describe the IESR entities are based on open standards where possible. Thus many are taken from the Dublin Core [6] namespace. Some properties are taken from RSLPCD, which is a 'de facto' rather than an 'official' standard for collection metadata. Work is currently in progress by the Dublin Core Metadata Initiative (DCMI) Collection Description working group to propose some collection properties within a Dublin Core namespace. If this is successful, IESR may migrate some of its RSLPCD terms to the new DCMI terms, but with consideration for backwards compatibility.

The IESR metadata properties are defined formally [7] as a Dublin Core

Application Profile [8]. An application profile provided a useful way to document all of the metadata properties and their corresponding namespaces. An additional field, 'searchable', indicates whether the value of a particular property is available for discovery, some properties being solely informational, and its corresponding search index attributes. The application profile also defines controlled vocabularies (encoding schemes) applicable to particular properties. The application profile uses some IESR-specific terms that are defined within an IESR namespace.

### **2.1 Identification of IESR Entities**

Every entity registered in IESR is assigned a unique global identifier. The form of identifier used is a PURL-based Object Identifier (POI) [9]. The POI convention provides a simple means of assigning 'relatively persistent' global identifiers within the Internet's 'http' namespace. They are unambiguous, being based on the IESR's internet domain. The remainder of the identifier is generated dynamically, based on the time and host machine process identifier, thus ensuring uniqueness. Within IESR metadata all entity identifiers and relation links use IESR POIs.

#### **Example 1. A Global Identifier for an IESR Entity**

<http://purl.org/poi/iesr.ac.uk/1056366559-25788>

### **2.2 Collection Metadata**

As discussed above, collections within IESR are described using a combination of Dublin Core and RSLPCD properties. Probably the significant properties for discovery are 'dc:title', 'dc:subject' and 'dcterms:abstract' (description).

Subject properties for collection metadata are limited to a small set of possible controlled vocabularies, mainly those endorsed by Dublin Core, but with the addition of those used widely within certain domains within UK academia. This restriction should enable quality results from distributed meta-searching. IESR requires at least one term from a single, common controlled vocabulary, specifically the Dewey Classification system [10], to further enhance discovery consistency when selecting collections by subject.

An IESR-specific property, ‘usesControlledVocabulary’, is introduced to indicate which controlled vocabularies are in use by a collection, values for this property being taken from an IESR-defined list. This property could provide data for a suggested terminology service that maps between subject schemes, and it provides information to portals as to whether they can search a particular collection using a particular controlled vocabulary.

A collection within IESR may have several, but must have at least one, services registered within IESR. A collection’s service is captured as an ‘iesr:hasService’ relation property. An IESR-specific property referring to the service entity seemed more appropriate than using the RSLPCD ‘locator’ property or the new term ‘isAvailableAt’ proposed by the DCMI Collections working group, the definitions of those covering physical as well as digital locations and not linking to further metadata.

Several further properties may record the coverage, geographic, temporal and educational of the collection, containing or associated collections, and related publications.

Information about rights and restrictions on using a collection is captured in several properties as free text statements. ‘dc:rights’ records any copyright statement about the collection. ‘iesr:useRights’ contains a statement about allowed usage of items from the collection, such as terms and conditions. ‘dcterms:accessRights’ holds information about any licence requirements to access the collection. The values of these properties pertain to the collection whatever method of access may be used. Further access restriction and authentication information is recorded for a collection’s particular services within the service metadata.

### Example 2. A Collection Description

```
<dc:type>Collection>
<dc:title>zetoc</dc:title>
<dc:identifier xsi:type="dcterms:URI">
http://purl.org/pt/iesr.ac.uk/1056366559-25788
</dc:identifier>
<dcterms:abstract>The zetoc database, the
British Library's ETOC, contains...
</dcterms:abstract>
<dc:type xsi:type="dcterms:DCMIType">
Collection</dc:type>
<dc:type xsi:type="rslpcd:CLDT">
Catalogue.Library.Text</dc:type>
<dc:rights>Copyright (c) British Library
1993-2004</dc:rights>
<iesr:useRights>All Rights Reserved.
http://zetoc.mimas.ac.uk/terms.html
</iesr:useRights>
<dcterms:accessRights>
Available conditionally free to UK FE and
HE. Available by subscription to...
</dcterms:accessRights>
<iesr:hasService xsi:type="dcterms:URI">
http://purl.org/pt/iesr.ac.uk/1056380019-18263
</iesr:hasService>
<dc:subject xsi:type="dcterms:DDC">
050</dc:subject>
<dc:subject xsi:type="dcterms:LCSH">
Medicine</dc:subject>
<rslpcd:contentsDateRange
xsi:type="dcterms:W3CDTF">
1993/</rslpcd:contentsDateRange>
<iesr:usesControlledList
```

```

    xsi:type="iesr:CtrlDVocabsList">
DDC</iesr:usesControlledList>
<rsrpcd:owner xsi:type="dcterms:URI">
http://purl.org/poi/iesr.ac.uk/1056381752-28099
</rsrpcd:owner>
<rsrpcd:hasPublication>
http://zetoc.mimas.ac.uk</rsrpcd:hasPublication>
</dcmltype:Collection>

```

### 2.3 Service Metadata

Previous implementations based on RSLPCD have used the 'locator' property to capture information about access to the collection, possibly the URL for a digital collection, or textual information for a physical collection. When designing the metadata for IESR it became apparent that more data than just an access URL would be required for a service. Most significant is the method of access, e.g. Z39.50 [11], SOAP [12], etc. Possibly this detail could be include as an XML attribute on the 'locator' element, but for some services further information may be needed, for example authentication requirements. Also IESR has a requirement to describe transactional services that do not have an associated collection, and thus to capture their title and description. Therefore IESR decided to describe a service as a separate entity with a set of 'service metadata'. There did not appear to be an existing standard schema to describe services for simple resource discovery, necessitating the definition of a bespoke IESR service metadata schema. However metadata properties from open standard schemes are used where possible. It is hoped this schema will provide a prototype definition for use by future projects.

In designing the service metadata, IESR recognised that its primary purpose is resource discovery, whereas providing information necessary to

connect to a service is of secondary importance. Thus IESR has a searchable set of metadata for a service including its location, with an additional property for some service types that details how to access a discovered service, including its possible arguments and its result formats. The value of this connection property is a by-reference pointer to a further set of metadata according to a schema that is appropriate for the particular service access method.

IESR service metadata includes a title and a description. 'Title' is probably redundant for a collection-based service, and thus is searchable for transactional services only. It is expected that the description would be used primarily for discovery of transactional services. However there may be cases where a particular service access point to a collection would benefit from some specific detail beyond the collection description.

The access URL for a service is captured using the RSLPCD property 'locator'. A service must have a single access point. For a Z39.50 service a Z39.50 Search URL (z3950s) [13] is used, which captures the port and database information as well as the host address.

A service has a single access method captured in a dc:type property according to an IESR controlled vocabulary. All common service access methods are recognised within IESR, including Z39.50, Web Services SOAP, SRW (Search – Retrieve – Web) [14], and OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) [15]. IESR is intended primarily to enable a portal to discover a collection of interest and then connect to it via its service. Thus the majority of service access methods are suitable for server-to-server

communication. However, in reality, for many collections currently available in the JISC Information Environment the only means of access is via a Web interface. Thus a pragmatic decision was made to include 'webpage' as an access method, realising that a portal could at least provide an end-user with a Web link to the collection. A further service type is 'webcgi', used to describe a service with a proprietary interface over HTTP CGI (Common Gateway Interface).

For some service types the location URL provides sufficient information to connect to the service. For example, given the access point to an OAI-PMH service, a portal can interrogate the service itself to discover further details such as which metadata formats it supports. But for other service types further details are required, thus suggesting the introduction of an IESR connection property 'interface'. For a SOAP service this property captures the address of the WSDL file. For a Z39.50 service `iesr:interface` points to a ZeeRex [16] file, an XML version of the Z39.50 'explain' information detailing the search attributes and result formats supported. IESR generates ZeeRex files on data supply for those services that do not already capture 'explain' details in this way. Proprietary 'webcgi' services posed a problem in capturing the argument keys and any fixed values required to connect to the service. Currently IESR generates a 'keys' file, using a bespoke IESR format, from data supply information for a 'webcgi' service, the `iesr:interface` property pointing to it.

IESR service metadata includes some minimal authentication information, just the style of authentication, e.g. IP address check, Athens [17], etc. It is

probable that this area will be extended in future to encompass authentication developments within the JISC Information Environment. A property is provided to capture the domain of use for a particular service, for example an OpenURL resolver's use would be relevant only to members of a particular institution. Both of these properties use `dcterms:accessRights` with appropriate encoding schemes.

A further property is `iesr:supportsStandard`. This captures details about which versions and profiles of a standard access method a service supports, for example a Z39.50 service may support 'level 1 functional area C' of the Bath Profile [18]. This property is repeatable, for example an OpenURL resolver may support version 0.1 OpenURL and the San Antonio Profile Level 1 of version 1.0 of the OpenURL Framework [19]. It was the possibility of a service supporting multiple versions or profiles of a standard that influenced the inclusion of this IESR-specific property rather than this detail being part of the access method. The list of access methods is deliberately kept as a simple list of access types, without any version information, to aid discovery and uncomplicated understanding by portals where further detail is not required.

The complete set of metadata to describe a service in IESR is defined in the Application Profile [7].

### Example 3. A Service Description

```
<dc:mitype:Service>
<dc:title>zetoc Z39.50 search</dc:title>
<dc:identifier xsi:type="dcterms:URI">
http://purl.org/poi/iesr.ac.uk/1056380019-18263
</dc:identifier>
<rs:locator xsi:type="dcterms:URI">
z3950s://zetoc.mimas.ac.uk:2121/zetoc
</rs:locator>
<iesr:interface xsi:type="dcterms:URI">
```

```

http://www.mimas.ac.uk/iesr/metadata/examples/interf
aces/svc-1056380019-18263-z.xml
</iesr:interface>
<dc:type xsi:type="iesr:AccMthdList">
z3950</dc:type>
<dcterms:accessRights xsi:type="iesr:AuthList">
ip</dcterms:accessRights>
<dcterms:accessRights xsi:type="iesr:AuthList">
athens</dcterms:accessRights>
<iesr:supportsStandard xsi:type="iesr:StdsList">
bath-1-c</iesr:supportsStandard>
<rslnpcd:seeAlso xsi:type="dcterms:URI">
http://zetoc.mimas.ac.uk/z3950.html
</rslnpcd:seeAlso>
<rslnpcd:administrator xsi:type="dcterms:URI">
http://purl.org/poi/iesr.ac.uk/1056381864-28646
</rslnpcd:administrator>
</dc:mitype:Service>

```

## 2.4 Agent Metadata

The only essential detail in the agent metadata, apart from the IESR-assigned identifier, is the single organisation name. Data suppliers are required to provide at least a contact email address for any agent that is an administrator. In the future IESR will include functionality to monitor the availability of services and would contact an administrator if a service were consistently unavailable.

### Example 4. An Agent Description

```

<iesr:Agent>
<dc:title>MIMAS</dc:title>
<dc:identifier xsi:type="dcterms:URI">
http://purl.org/poi/iesr.ac.uk/1056381864-28646
</dc:identifier>
<iesr:email>info@mimas.ac.uk</iesr:email>
<iesr:phone>+441612756109</iesr:phone>
<dc:relation xsi:type="dcterms:URI">
http://www.mimas.ac.uk</dc:relation>
</iesr:Agent>

```

## 2.5 Administrative Metadata

Every entity in the IESR includes a set of administrative metadata within a containing element, `iesr:admata`. Data suppliers may include details of who created the metadata record and when. A `dcterms:modified` field is set automatically by IESR when a metadata

record is registered or updated, this information being used by the IESR when providing OAI-PMH data for harvesting.

A `dc:rights` property defines the restrictions for using the IESR metadata record. All metadata in IESR is freely available for non-commercial use under a Creative Commons [20] licence as long as attribution of provenance and the same licence are maintained (non-commercial, share-alike, attribution required). The supply of data to IESR by an organisation implicitly indicates that they agree to the licence.

## 3 IESR XML Records

### 3.1 Data Within IESR

As described above, the IESR data model comprises three separate entities with various relationships between them, for example an agent may be the owner of several collections and the administrator of several services. However, the current implementation of IESR, described further in section 4, requires a flat XML data structure, with a single XML record for a discoverable item. Thus, within IESR, data is held as composite records for collections and transactional services, rather than as separate individual entities. A composite collection record in IESR includes: the collection metadata; the metadata for all the services that provide access to it; the metadata for its owner agents; and the metadata for the administrative agents for the services that provide access to it.

This composite data structure within IESR obviously results in multiple copies of some entity records, in particular the agents. But the extra file store required, not a significant issue on the current platform, is an expense

worth paying to simplify the IESR discovery implementation. The composite records also introduce maintenance concerns. If an agent is updated all the composite collection records that include it must be updated. This problem is resolved by the introduction of the IESR Meta-Registry described below in section 3.3.

### **3.2 Metadata External to IESR**

IESR metadata records retrieved by Z39.50 or harvested via OAI-PMH are composite records as described above. However, data is supplied to IESR, either new or updated records, as separate entities.

### **3.3 The IESR Meta-Registry**

The IESR Meta-Registry was introduced, partly to provide a browsing interface to IESR, and partly to overcome the maintenance issues described above. The Meta-Registry holds simple details of all the records registered with IESR including an identifier and a title. For each entity it records relationships, that is 'has service' and 'owner' identifiers for a collection and 'administrator' identifiers for a service.

When an entity is registered with IESR it is assigned an identifier, if it is a new entity, and a record is created for it in the Meta-Registry. The 'base' entity description is kept separately, stripped of its relation properties. After each registration the Meta-Registry is updated to set any relation links to the new record. The IESR database is built from the data in the Meta-Registry and the 'base' registered entity records.

## **4 The Registry**

### **4.1 Implementation**

The registry is implemented using Cheshire II [21], which is a next generation online catalogue and full text information retrieval system, developed using advanced information retrieval techniques. It is open source software, free for non-commercial uses, and was developed at the University of California-Berkeley School of Information Management and Systems, with later development also at the University of Liverpool. Cheshire is the chosen platform because of its powerful discovery capability, coupled with its integral Z39.50 and primitive Web interfaces, and also because of existing expertise within the project team [22] [23]. The data is held within the Cheshire database as XML records, the flat, composite records described above, along with a set of Cheshire indexes to support discovery and retrieval.

The IESR database is built overnight whenever a record has been added to, or updated in, the Meta-Registry. Rebuilding the entire database every time may cause performance issues if the IESR becomes large, so this decision will be readdressed in the future. But it is the chosen process for the prototype to overcome data maintenance problems and for rapid development.

### **4.2 IESR Interfaces**

IESR provides several interfaces to allow both portals and humans to interrogate the data, and more, including SRW, are planned.

#### **4.2.1 The IESR Web Interface**

The Web interface is provided mainly for content checking by data suppliers,

IESR being primarily a server-to-server application. The Web interface could also be used by portal personnel to make manual decisions on the inclusion of collections. It could potentially be used for general resource discovery within the JISC Information Environment and by JISC collection managers.

#### **4.2.2 The IESR Z39.50 Interface**

IESR has a Z39.50 interface able to supply records as SUTRS (Simple Unstructured Text), simple Dublin Core and IESR XML. SUTRS records are the same as the full records from the Web interface but formatted as plain text.

Each simple Dublin Core result in the returned list consists of a set of descriptions one for each entity within the composite collection record. `dc:relation` properties link between these descriptions. These simple Dublin Core descriptions lose much of the richness of the IESR data, but are provided to satisfy a requirement to support the Bath Profile.

IESR XML may be retrieved by requesting XML with an IESR 'element set'. The returned results are composite IESR XML records as described above in section 3.

#### **4.2.3 The IESR OAI-PMH Interface**

Work is currently underway to provide an OAI-PMH interface to enable harvesting of IESR records. As required by the protocol, this interface will supply a simple Dublin Core record. Unlike the Z39.50 simple Dublin Core results described above, this is a single Dublin Core record, OAI-PMH not including the possibility of a composite result. This single Dublin Core record will cause more severe 'dumbing down', making the consequent

simplified record of little obvious use. It would be expected that most servers that harvest IESR data would want the rich full IESR descriptions. There appear to be two solutions to this dilemma, with the possibility of both being provided.

Firstly, IESR could invent a new, proprietary result format for OAI-PMH harvesting that will return complete IESR records. The advantage of this approach is that it provides simple retrieval of full IESR records for applications that understand IESR data and know the proprietary token for IESR XML retrieval. It should be remembered that currently IESR is a service within the 'closed' JISC Information Environment domain, but it would not be sensible to provide an interface that would deter wider use. The disadvantage of this solution is that IESR would be providing a non-standard OAI-PMH interface not understood by harvesting servers in general. This disadvantage may not be significant, it being likely that a proprietary format would be ignored by a server outside of the IESR domain.

The second solution would be to provide a very simple Dublin Core record including just the salient details of a collection, but with a 'by-reference' link, a pointer within a `dc:relation` property, to a full IESR record. The advantage of this approach is that it maintains a standard OAI-PMH interface and it provides basic Dublin Core records to servers that require no further details. The disadvantage is that a server that requires a full XML record would have to make a second retrieval to obtain it.

If the second approach were implemented the retrieval link for a full IESR record could be an OpenURL. OpenURL was developed as a standard

way of passing information about a resource between a source application and an OpenURL-aware resolver [24]. Its original and primary purpose is to enable a researcher to link from a referenced article to a full text copy of that article where the researcher's institution has a valid subscription. During the process of proposing the OpenURL Framework as a NISO standard, Z39.88-2004, other possible uses of OpenURL were envisaged including server-to-server communication.

A possible OpenURL for a 'by reference' link is given in Example 5. Note that in this example a hypothetical resolver address is used. An actual OpenURL would be 'URL escape encoded', with special characters in hexadecimal format for safe HTTP transmission, but this encoding has been omitted, and line-breaks have been added to the OpenURL, for readability. Within the OpenURL, the IESR record requested, the referent in OpenURL terminology, is described by an IESR global identifier. The experimental Dublin Core metadata format is used to request a service that returns an XML record for a particular entity type.

**Example 5. An OpenURL to Retrieve an IESR XML Record**

```
http://iesr.ac.uk/ourllinkto?  
url_ver=Z39.88-2004  
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx  
&rft_id=  
http://purl.org/pci/iesr.ac.uk/1056366559-25788  
&svc_val_fmt=info:ofi/fmt:kev:mtx:dc  
&svc.format=text/xml  
&svc.type=Service
```

## 5 Discussion and Conclusion

IESR is currently a prototype holding the first set of data supplied by the main data centres and services within the JISC Information Environment. It still

has to prove itself as a shared service in use by portals. There is much work still to be done to develop further interfaces for data supply and retrieval.

From the first tranche of data supply it has become apparent that many collections of interest to researchers, learners and teachers within the JISC Information Environment have only a Web interface and so are currently unsuitable for use within a server-to-server environment. It appears that some encouragement will be needed for administrators to provide more functional interfaces to their collections before the full potential of a shared-service registry will be achieved. However the IESR as a central repository of collections and services within the JISC Information Environment is still a viable goal. The Web interface to a collection can be used by a portal to provide to an end-user a link to a collection, even though it cannot include the collection in a distributed search.

The choice of a Cheshire platform, whose underlying data model, a flat XML record, does not match the IESR data model of linked entities, has created some implementation issues. However these problems seem to have been overcome successfully and no consequent implementation concerns are yet apparent.

Designing the metadata for IESR was an important task, because its goal was to define an interoperable, stable set of metadata based on existing open standards. Although an early decision was made to use RSLPCD, decisions had to be made about which properties were extraneous to records describing digital collections primarily for discovery. Designing the service metadata was more significant because

there did not appear to be any existing models to follow for simple resource discovery. Hopefully the IESR service metadata will provide a paradigm for future similar applications.

### Acknowledgements

The IESR prototype development was funded by the Joint Information Systems Committee (JISC) [25] of the UK Higher and Further Education Councils as part of its ‘Shared Services’ programme. IESR is hosted by MIMAS [26] at the University of Manchester. The author wishes to acknowledge the assistance of colleagues in the development of the IESR metadata, in particular Pete Johnston and Andy Powell of UKOLN [27] at the University of Bath, but also other members of the IESR project team who contributed: Amanda Hill, the project manager, and Leigh Morris, the content manager, of MIMAS; and Amanda Closier, who gathered requirements from potential stakeholders, and Rachel Heery of UKOLN.

### References

1. The JISC Information Environment Service Registry. <http://www.iesr.ac.uk/>
2. NISO Committee AX and A. Apps. Z39.88-2004: The Key/Encoded Value Format Implementation Guidelines [http://www.openurl.info/registry/docs/implementation\\_guidelines](http://www.openurl.info/registry/docs/implementation_guidelines)
3. The JISC Information Environment. [http://www.jisc.ac.uk/index.cfm?name=ie\\_home](http://www.jisc.ac.uk/index.cfm?name=ie_home)
4. RSLP Collection Description. <http://www.ukoln.ac.uk/metadata/rsrp/schema/>
5. P. Johnston. The JISC Information Environment Service Registry. 2004. To be available via <http://www.ukoln.ac.uk/cd-focus/briefings/>
6. The Dublin Core Metadata Initiative. <http://www.dublincore.org>
7. A. Apps, P. Johnston and A. Powell. IESR Application Profile. 2004. <http://www.iesr.ac.uk/profile/>
8. CEN/ISSS CWA 14855, Dublin Core Application Profile Guidelines. 2003. <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/MMI-DC/cwa14855-00-2003-Nov.pdf>
9. A. Powell, J. Young, and T. Hickey. PURL-based Object Identifier (POI). <http://www.ukoln.ac.uk/distributed-systems/poi/>
10. The Dewey Decimal Classification System. <http://www.oclc.org/dewey/index.htm>
11. Z39.50, for information retrieval. <http://www.niso.org/standards/resources/z3950.pdf>
12. W3C Web Services SOAP. <http://www.w3.org/2000/xp/Group/>
13. z3950s: the Z39.50 Search URI. <http://www.ietf.org/rfc/rfc2056.txt>
14. SRW: Search – Retrieve – Web. <http://www.loc.gov/z3950/agency/zing/srw/>
15. Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). [http://www.openarchives.org/OAI/openarchives\\_protocol.html](http://www.openarchives.org/OAI/openarchives_protocol.html)
16. ZeeRex: The Explainable “Explain” Service. <http://explain.z3950.org>
17. Athens Access Management System. <http://www.athens.ac.uk>
18. The Z39.50 Bath Profile. <http://www.nlc-bnc.ca/bath/ap-bath-e.htm>
19. Z39.88-2004, The OpenURL Framework for Context-Sensitive Services. Available via <http://library.caltech.edu/openurl/Standard.htm>
20. Creative Commons. <http://creativecommons.org>
21. Cheshire II. <http://cheshire.lib.berkeley.edu>
22. A. Apps, R. MacIntyre and L. Morris. Exposing Cross-Domain Resources for Researchers and Learners. In: Proc. DC2002: Dublin Core and Metadata for e-Communities, DC2002. Firenze University Press, 2002 pp71-80, ISBN 88-8453-043-1. <http://epub.mimas.ac.uk/papers/mimasdc2002.html>
23. A. Apps and R. MacIntyre. Prototyping Digital Library Technologies in zetoc. Lecture Notes in Computer Science 2458: 309-323, 2002. <http://epub.mimas.ac.uk/papers/appsmacecdl2002.html>
24. H. Van de Sompel and O. Beit-Arie. Open Linking in the Scholarly Information Environment Using the OpenURL Framework. D-Lib Magazine. 7(3), 2001.
25. JISC. <http://www.jisc.ac.uk>
26. MIMAS. <http://www.mimas.ac.uk>
27. UKOLN. <http://www.ukoln.ac.uk>