

# *OpenURL – The Rough Guide*

Tony Hammond  
Nature Publishing Group

# *OpenURL – One Perspective*

*"The person publishing the data will feel that they have the right to tell it how to look. But that's just one side of it. It's the [user] who's really in control."*

*– Tim Berners-Lee,  
WWW2004, May 19, 2004*

# *OpenURL – Link Interrupted*

- ◆ A confession – this is what I was originally tempted to entitle this presentation
- ◆ Because this is really the whole **essence** of **OpenURL**
- ◆ Interrupting – or better **mediating** – Web links

# *OpenURL – A Loose Definition*

*"OpenURL is a generic Web technology that enables the context of a service request to be evaluated within the user environment (i.e. the user, or rather the user's institution, is allowed a determination of the link endpoints)."*

*– (an unofficial definition)*

# *OpenURL – Notes (1)*

## ◆ First Note:

There are  
two flavours  
of  
OpenURL

- ◆ 1) The special theory, i.e. the de facto SFX standard
- ◆ 2) The general theory, i.e. the ANSI/NISO Z39.88-2004 (draft) information standard

# *OpenURL – Notes (2)*

## ◆ Second Note:

OpenURL talks to **applications** not to authorities – it is therefore truly an **open** technology

◆ Hence, **link servers** and friends

◆ The overwhelming majority of links on the Web are **proprietary** and have limited utility

◆ OpenURL links by contrast can be **recycled** through any OpenURL-aware service

# *OpenURL – Notes (3)*

## ◆ Third Note:

OpenURL links are distinguished URIs

◆ They stand out in contrast to the majority of URIs which are opaque

◆ They are immediately recognizable by a brand self-declaration

◆ They can be parsed against a data model – and so proven to be true

# *OpenURL – How it Stacks Up*

## ◆ Usual Web links:

- ✓ Proprietary
- ✓ Authoritative
- ✓ Narrowband
- ✓ Assured
- ✓ Single-use
- ✓ Opaque

## ◆ OpenURL links:

- ✓ Public
- ✓ Productive
- ✓ Broadband
- ✓ Non-assured
- ✓ Repurposable
- ✓ Non-opaque

# *OpenURL – The Big Ideas*

- ◆ User is a participant in the linking process
- ◆ Describe the resource don't just blindly point at the resource
- ◆ So, put the user in control
- ◆ Send public, and open descriptions
- ◆ And, return context-sensitive services for the user
- ◆ Just-in-time links not just-in-case links

# *OpenURL – History (1)*

- ◆ 1999 – Early work at Ghent University on linking **heterogenous** collections within their institution
- ◆ This linking prototype provided both **relevant** and **extended** services
- ◆ And because of the perceived magic this was dubbed **SFX** (simple moniker for *Special Effects*)

# *OpenURL – History (2)*

- ◆ Meanwhile ANSI/NISO was looking into the **Appropriate Copy** problem
- ◆ When a library has multiple copies of an object, **which copy is right** for the patron?
- ◆ **Appropriate Copy** is a subset of the more general problem of provisioning **extended services**
- ◆ In 1999 Ghent University partnered a cross-institution experiment with Los Alamos

# *OpenURL – History (3)*

- ◆ In 2000 Caltech started up an SFX service and released this to their patrons in 2001
- ◆ Other SFX-style services and installations were quickly introduced
- ◆ Early 2000 Ghent University signed over the rights to SFX to Ex Libris
- ◆ Mid 2004 the total number of OpenURL installations is growing towards the 1000 mark

# *OpenURL – Link Server (1)*

- ◆ Consider the following journal citation:

*Paul Smith, Science,  
12 (1998) 3, 1–8.*

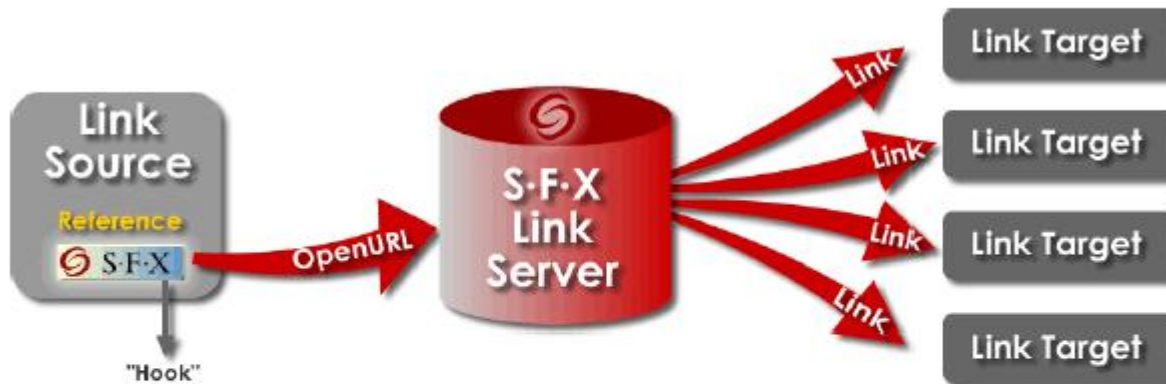
- ◆ It contains several metadata elements

- ◆ The SFX spec for OpenURL supports several common bibliographic elements for journals, books, etc.

# OpenURL – Link Server (2)

| <i>Key</i>                  | <i>Value</i> |
|-----------------------------|--------------|
| genre (content type)        | article      |
| issn                        | 0036-8075    |
| volume                      | 12           |
| issue                       | 3            |
| spage (start page)          | 1            |
| epage (end page)            | 8            |
| date                        | 1998         |
| aulast (author last name)   | Smith        |
| aufirst (author first name) | Paul         |

# OpenURL – Link Server (3)



[http://sfx.aaa.edu/menu?genre=article&issn=0036-8075  
&volume=12&issue=3&spage=1&epage=8&date=1998  
&aulast=Smith&aufirst=Paul](http://sfx.aaa.edu/menu?genre=article&issn=0036-8075&volume=12&issue=3&spage=1&epage=8&date=1998&aulast=Smith&aufirst=Paul)

Slide provided courtesy of Ex-libris

# OpenURL – Link Server (4)

The screenshot illustrates an OpenURL workflow. On the left, the 'SFX Source' is a metadata record for a journal article, with the SFX logo highlighted. In the center, the 'SFX Menu' is a list of actions such as 'Full text available from', 'Request document via', and 'Find this author's email in'. On the right, the 'SFX Target' is the resulting article page on PNASOnline, titled 'Color vision: Opsins and options' by J. D. Mollon.

**SFX Source** → **SFX Menu** → **SFX Target**

Slide provided courtesy of Ex-libris

# *OpenURL – History (4)*

- ◆ End 2000 the SFX spec was submitted to NISO as a candidate standard
- ◆ Early 2001 NISO committee AX was established to fast-track standardization of OpenURL
- ◆ Preparatory work revealed a latent **data structure** behind the everyday use of OpenURL
- ◆ That data structure was termed a **ContextObject**

# *OpenURL – ContextObject*

◆ The `ContextObject` is simply a container of `Entity` descriptions

◆ These `Entities` are just `Resources` that can be described

◆ `Resources` are the

– *What?*

– *Where?*

– *Whence?*

– *Who?*

– *How?*

– *Which?*

of the user request

# OpenURL – Entities

◆ Referent (*What?*)

◆ Referring-Entity  
(*Where?*)

◆ Referrer (*Whence?*)

◆ Requester (*Who?*)

◆ Service-Type (*How?*)

◆ Resolver (*Which?*)

# *OpenURL – Descriptors*

◆ Entities are described by means of Descriptors

◆ Descriptors are:  
a) Public  
b) Private

◆ Descriptors are:

- a) Identifiers
- b) Metadata
  - By Reference
  - By Value
- c) Private Data

# *OpenURL – Identifiers*

◆ Prior to the release of the **Draft Standard for Trial Use** in 2003, the draft defined three naming systems:

**URI** – 'church' names

**ORI** – 'street' names

**XRI** – 'house' names

◆ Following feedback received, NISO AX decided to move to an all-URI approach

◆ This led to the development of a new URI scheme – **INFO**

# *OpenURL – Formats (1)*

◆ The concept of a **Format** was elaborated

◆ A Format is defined as a **triple** with the following components:

◆ 1) Representation

◆ 2) Constraint Language

◆ 3) Constraint Model (or Schema)

# *OpenURL – Formats (2)*

◆ Two Representations of the ContextObject are defined:

a) KEV

b) XML

◆ A KEV (or Key/Encoded Value) Representation can be sent inline on an OpenURL querystring

◆ An XML Representation can be pointed at by the OpenURL or sent in a data POST

# *OpenURL – Formats (3)*

◆ The draft standard defines four metadata schemas:

- 1) Book
- 2) Journal
- 3) Patent
- 4) Dissertation

◆ These are available in both KEV and XML representations

◆ Also available for trial use is a [Dublin Core Format \(KEV\)](#)

# *OpenURL – The Registry (1)*

◆ To disclose or publish Identifiers, Metadata Formats, and other core components, a Registry mechanism was established

◆ The draft standard – now termed the OpenURL Framework – makes available two key deliverables: 1) a data model – the ContextObject, and 2) the Registry

# *OpenURL – The Registry (2)*



[The NISO AX Committee for the OpenURL](#)

**Registry for the OpenURL Framework - ANSI/NISO Z39.88-2004**

[Repository Identification](#) | [Registry Entries](#) | [Implementation Guidelines](#)

## **Components of OpenURL Framework**

[Namespaces](#)

[Character Encodings](#)

[Physical Representations](#)

[Constraint Languages](#)

[ContextObject Formats](#)

[Metadata Formats](#)

[Transports](#)

[Community Profiles](#)



# OpenURL – The Registry (3)



[The NISO AX Committee for the OpenURL](#)

[Registry for the OpenURL Framework - ANSI/NISO Z39.88-2004](#)

[Repository Identification](#) | [Registry Entries](#) | [Implementation Guidelines](#)

## Core:Namespaces

|  |   |
|--|---|
| <a href="#">info:ofi/</a>                  | Namespace reserved for OpenURL Framework Registry Identifiers   |
| <a href="#">info:ofi/enc:</a>              | Namespace reserved for Registry Identifiers of Character Encodings  |
| <a href="#">info:ofi/fmt:</a>              | Namespace reserved for the identification of ContextObject Formats, Metadata Formats, Physical Representations, and Con |
| <a href="#">info:ofi/nam:</a>              | Namespace reserved for Registry Identifiers of Namespaces.  |
| <a href="#">info:ofi/nam:data:</a>         | Namespace for "data" URI Scheme   |
| <a href="#">info:ofi/nam:ftp:</a>          | Namespace for "ftp" URI Scheme  |
| <a href="#">info:ofi/nam:http:</a>         | Namespace for "http" URI Scheme   |
| <a href="#">info:ofi/nam:https:</a>        | Namespace for "https" URI Scheme  |
| <a href="#">info:ofi/nam:info:</a>         | Namespace for "info" URI Scheme   |
| <a href="#">info:ofi/nam:info:bibcode:</a> | Namespace of Astrophysics Bibcodes  |
| <a href="#">info:ofi/nam:info:doi:</a>     | Namespace of Digital Object Identifiers   |
| <a href="#">info:ofi/nam:info:hdl:</a>     | Namespace for CNRI handles  |
| <a href="#">info:ofi/nam:info:lccn:</a>    | Namespace of Library of Congress Control Numbers  |
| <a href="#">info:ofi/nam:info:oai:</a>     | Namespace of OAI Identifiers  |
| <a href="#">info:ofi/nam:info:oclcnum:</a> | Namespace of identifiers assigned by OCLC to records in the WorldCat database   |
| <a href="#">info:ofi/nam:info:pmid:</a>    | Namespace of PubMed Identifiers   |
| <a href="#">info:ofi/nam:info:sici:</a>    | Namespace of SICI Codes   |
| <a href="#">info:ofi/nam:info:sid:</a>     | Namespace for identifiers that follow the info:sid scheme   |
| <a href="#">info:ofi/nam:ldap:</a>         | Namespace for "ldap" URI Scheme   |
| <a href="#">info:ofi/nam:mailto:</a>       | Namespace for "mailto" URI Scheme   |
| <a href="#">info:ofi/nam:urn:</a>          | Namespace for "urn" URI Scheme  |

# *OpenURL – Transports*

- ◆ Multiple transport mechanisms can be registered
- ◆ Currently two methods are defined:
  - a) HTTP
  - b) HTTPS
- ◆ Transporting a `ContextObject` (whether by value or by reference) over either of these transports is termed an `OpenURL`

# *OpenURL – Profiles*

- ◆ Profiles allow communities to select core components from the Registry

- ◆ Enables vendor compliance

- ◆ The draft standard defines two profiles:

- 1) San Antonio Level 1 (KEV)
- 2) San Antonio Level 2 (XML)

- \* Also a Dublin Core profile (KEV)

# *OpenURL – Example (1)*

# Referent

rft\_id = info:pmid/11823195

# Referring-Entity

rfe\_val\_fmt = info:ofi/fmt:opl:bnf:journal

rfe.issn = 1234-5678

rfe.volume = 68

rfe.issue = 2

rfe.date = 2002-02

## *OpenURL – Example (2)*

&rft\_id=info:pmid/11823195&rfe\_val\_fmt=info:  
ofi/fmt:kev:mtx:journal&rfe.issn=1234-  
5678&rfe.volume=68&rfe.issue=2&rfe.date=  
2002-02

```
<ctx:context-object ctx:version="Z39.88-2004" >
```

```
<ctx:referent>
```

```
<ctx:identifier>info:pmid/11823195</ctx:identifier>
```

```
</ctx:referent>
```

```
<ctx:referring-entity>
```

```
<ctx:metadata-by-val>
```

```
<ctx:format>info:ofi/fmt:xml:xsd:journal</ctx:format>
```

```
<ctx:metadata>
```

```
<j:journal
```

```
xmlns:j="http://www.openurl.info/ori/fmt/xml/xsd:"
```

```
<j:issn>1234-5678</j:issn>
```

```
<j:volume>68</j:volume>
```

```
<j:issue>2</j:issue>
```

```
<j:date>2002-02</j:date>
```

```
</j:journal>
```

```
</ctx:metadata>
```

## OpenURL – Example (3)

# *OpenURL – Use Cases*

- ◆ Generic service query API
- ◆ RSS 1.0 module `mod_context`
- ◆ Interface to portal cluster
- ◆ Doc del service to query holdings including print
- ◆ Cross-platform shopping cart service

# *OpenURL – Purpose*

◆ And what is the point of OpenURL in the new reformulation?

◆ The OpenURL is used to send a ContextObject to an intermediary Service which performs some Magic\* for the user

\* The term Magic is not defined