

OpenURL-Aware ETDs

Eric F. Van de Velde and Betsy Coles

One of the significant advantages of ETDs over printed theses is the ability for incorporating active web links in ETDs. OpenURL technology amplifies this advantage by increasing the power of these web links. Already, an OpenURL-aware ETD may offer the following features:

1. Link each citation in the bibliography to a menu of services appropriate to the citation AND to the reader of the ETD. For example, a citation to a journal article might be linked to:
 - The full text of the article (if the reader has licensed access to the full text)
 - Document-fulfillment services that the reader is allowed to use
 - Online catalogs of libraries appropriate to the reader
 - Abstracting and indexing services in which the article may be cited and where the reader may find additional related literature.
2. Increase the longevity of the links by decreasing the reliance on static URLs, using metadata to describe the cited works instead.

The current state of the art is only the beginning. We expect OpenURL technology to expand beyond the world of bibliographic citations. It could be used to increase the functionality of subject headings and of references to chemical formulas, genomic sequences, or products. For example, a biology ETD on a particular gene might be linked to all other scholarly works that refer to this gene, whether or not the scholarly works existed at the time of writing the ETD, whether or not the author was aware of these other scholarly works. However, before OpenURL can be used for these applications, significant additional development is required. For now, we focus our attention on the application of OpenURLs to bibliographic citations.

What is an OpenURL?

A bibliographic citation describes a **referent**, which may be a journal article, a book, a technical report, or some other work. When the citation to the referent occurs within an electronic document like an ETD, it makes sense to embed the URL of the referent in the ETD so that the referent is only a mouse click away from the citation. In OpenURL terminology, the person who clicks on that link is called the **requester**. In an ETD context, any reader of the ETD is a potential requester.

Embedding URLs has some disadvantages, however. For example:

- The URL becomes effectively useless when the location of the referent changes.
- Not all referents have URLs. For example, non-digital works do not have a URL.
- Some referents are associated with more than one URL, only one of which may be appropriate for the requester. For example, if the referent is part of a licensed resource, requesters need the URL of that version to which they have access.

- Requesters may be interested in more than just the referent: they may want services related to the referent. For example, they may want to check whether a book is available in their institute libraries.

In the OpenURL approach, an indirect link replaces the direct link from citation to referent. The appropriately formatted citation is transported to a **resolver**, which transforms the citation into one or more URLs and/or into a menu of services. Because the resolver performs this transformation when the ETD is read, it is able to use information that was not known by the author of the ETD. This includes the identity of the requester, current URLs of referents, and scholarly information produced since the ETD was written. To implement this basic idea, we must

1. Cast the citation into a **format** that can be parsed by an automated service. This machine-readable format organizes the **metadata** obtained from the citation and from the context in which this citation occurs.
2. **Transport** the metadata via the web to the resolver.
3. Build a resolver that transforms metadata into services and/or URLs.

An OpenURL is a web-transportable metadata format. It is only concerned with steps 1 and 2 of the above process. While OpenURL is an enabling technology for linking services to citations, it is not concerned with the nature of these services or with the methods by which the metadata contained in the OpenURL are transformed into services. That belongs in the realm of resolvers. Whereas resolvers can be proprietary and closed systems, it is expected that OpenURL will become an open standard.

Draft guidelines for constructing OpenURLs are already freely available [[Van de Sompel, Hochstenbach, and Beit-Arie 2000](#)], and a formal standardization process has started under the aegis of NISO [[NISO Committee AX](#)].

The number of available OpenURL resolvers is growing rapidly. Currently, they include:

- SFX [[Ex Libris \(USA\), Inc. SFX](#)] was the first OpenURL resolver. In fact, the SFX resolver predates OpenURL. Van de Sompel and Hochstenbach developed the SFX resolver and the OpenURL concepts as part of their research on context-sensitive linking [[Van de Sompel and Hochstenbach 1999a, 1999b, and 1999c](#)].
- 1Cate, [jake.openly.com](#), and [link.openly.com](#) [[Openly Informatics, Inc.](#)]
- LinkFinderPlus [[Endeavor Information Systems, Inc.](#)]
- Open Linking Technology [[Fretwell-Downing, Inc.](#)]
- Powell's OpenResolver [[Powell](#)] is available under the GNU open-source license.

Demonstration and Technical Details

Consider the following citation:

Van de Sompel, Herbert and Beit-Arie, Oren. 2001. Open Linking in the Scholarly Information Environment Using the OpenURL Framework. D-Lib Magazine. 7(3).

Example 1: A typical conventional citation to a journal article

Using the draft OpenURL specifications [[Van de Sompel, Hochstenbach, and Beit-Arie 2000](#)], the OpenURL version of this citation could take the form displayed in Example 2.

<http://sfx.caltech.edu:8088/caltech?genre=article&atitle=Open%20Linking%20in%20the%20Scholarly%20Information%20Environment%20Using%20the%20OpenURL%20Framework&title=D-Lib%20Magazine&issn=1082-9873&date=2001-03&volume=7&issue=3&aulast=Van%20de%20Sompel>

Example 2: The citation of Example 1 formatted as an HTTP-encoded OpenURL

In other words, an OpenURL can take the form of a familiar HTTP GET and/or HTTP POST request. The part before the question mark is the URL of Caltech's SFX resolver [[Ex Libris \(USA\), Inc. SFX](#)]. The part following the question mark is the metadata describing the referent. In other words, this part is nothing but the citation in machine-readable form.

When the requester clicks on the above link, the requester's browser jumps to the URL of the resolver, and the metadata is transported to the resolver. What the resolver does with this information is not standardized in any way: resolver behavior is limited only by the imagination of resolver developers. In our example, the resolver produces a list of services appropriate to this particular citation. Figure 1 displays the list of services produced by the Caltech SFX resolver at the time of writing the current document. (Since the resolver and the database behind the resolver change over time, the list of services changes over time.)

SFX Services for this record

Title Open Linking in the Scholarly Information Environment Using the OpenURL Framework
Source D-LIB MAGAZINE [1082-9873]
 yr: 2001 vol: 7 iss: 3

Get Caltech-licensed full-text from **D-Lib Magazine**
 year: issue:

Check holdings in **Caltech's Library catalog (CLAS)**

Request delivery via **IBID**

Search author in **Web of Science**
 Name:

Lookup as cited author in **Web of Science**
 Name:

Lookup as cited reference in **Web of Science**
 Citation: Van de Sompel 2001 ...

Send feedback to **Caltech Library System staff**

Get the **Caltech SFX FAQ**

Search the **Web**

 Search:

© 2001 SFX by Ex Libris (USA) Inc.

Figure 1: List of services produced by Caltech's SFX resolver with the metadata of Examples 1 or 2 as input

The mechanism as explained thus far is inadequate. For example:

- Non-Caltech requesters are referred to Caltech resources, such as the document-delivery system Ibid or the catalog of the Caltech Library System.
- Documents containing links like those in Example 2 must be updated every time the URL of the resolver changes.

The resolver should be determined at the time when the requester clicks on the link. This can be achieved in several ways. Unfortunately, elegant solutions require web-browser modification, and we cannot wait for that to happen. For now, we must settle for pragmatic approaches, each of which has some drawbacks. For example, the URL of the resolver could be stored in a user profile. This works well for systems that require users to log in, but it does not work for ETD collections that are free and open to the public. In this case, one may have to resort (somewhat reluctantly) to web-browser cookies. This is not the proper forum to examine all possible approaches to resolver selection. However, it is instructive to examine the Cookie Pusher mechanism, first proposed by Van de Sompel and Hochstenbach [Van de Sompel and Hochstenbach 2000].

Before they can use the resolver, requesters must browse to a particular web page in order to set a cookie that contains the URL of the resolver. This one visit activates their access to the resolver until the cookie is deleted. If this cookie is not set, the data provider (in our case, the ETD collection) assumes the requester does not have access to

an OpenURL resolver and either does not provide resolver functionality or (if available) uses a free resolver that may be used by anyone.

For simplicity, we assume that the cookie has been set and that the ETD is formatted in HTML. Since we have no prior knowledge of the URL of the resolver, it is impossible to embed in the ETD an HTTP link like the one of Example 2. Instead, we have to retrieve the cookie, construct the HTTP request, and process the HTTP request. In an HTML-formatted ETD, this activity is “hidden” behind a button placed next to the citation as in Example 3.

Van de Sompel, Herbert and Beit-Arie, Oren. 2001. Open Linking in the Scholarly Information Environment Using the OpenURL Framework. D-Lib Magazine. 7(3). 

Example 3: An “OpenURL Aware” citation to a journal article

Typically, a program written in a browser-compatible language such as Java or JavaScript performs all of the actions required. Ex Libris (USA), Inc. [\[Ex Libris \(USA\), Inc. Script\]](#) provides such a script for the SFX environment. The script reads the user’s cookie, determines the appropriate resolver, and defines an SFXButton function that can be invoked in an HTML page. If the ETD includes this JavaScript, then the OpenURL-Aware citation of Example 3 can be encoded in HTML as shown in Example 4.

```
<P>Van de Sompel, Herbert and Beit-Arie, Oren. 2001. Open Linking in the
Scholarly Information Environment Using the OpenURL Framework. D-Lib
Magazine. 7(3).
<SCRIPT TYPE="text/javascript" >
SFXButton(
    "genre=article&
    atitle=Open%20Linking%20in%20the%20Scholarly%20Information%20Envir
onment%20Using%20the%20OpenURL%20Framework&
    title=D-Lib%20Magazine&issn=1082-9873&
    date=2001&
    volume=7&
    issue=3&
    aulast=Van%20de%20Sompel&
    auinit=H" )
</SCRIPT></P>
```

Example 4: HTML Representation of an OpenURL-Aware citation.

With this script in place, requesters who click on the “SFX button” are directed to the appropriate resolver and receive a tailored menu of services.

The bibliography of the online version of this Chapter [\[Van de Velde and Coles\]](#) shows this technique in action. For demonstration purposes, requesters without a cookie are given access to the Caltech SFX resolver. (This works for this particular online bibliography only. Access to the resolver may be temporary. Only Caltech-affiliated requesters have access to the services offered in the SFX menu.)

OpenURL Standardization

In February 2001, NISO formed NISO Committee AX and started the OpenURL standardization process. At the time of writing this, no NISO membership votes had been taken. What follows is an outline of the status of discussions within NISO Committee AX around the middle of January 2002 and is not endorsed by NISO or the NISO membership. For recent updates on the OpenURL standardization process, please check the NISO web site [\[NISO\]](#) or the NISO Committee AX web site [\[NISO Committee AX\]](#).

The Committee adopted both a short-term and a long-term approach. In the short time, it wanted to encourage early adoption of OpenURL by assuring reasonable stability to early adopters. In addition, the committee recognized that the OpenURL guidelines [\[Van de Sompel, Hochstenbach, and Beit-Arie 2000\]](#) are a great success both in number of early adopters and the quality of the applications. The committee, therefore, recommended this draft without amendments or modifications as Version 0.1 of the OpenURL standard. OpenURLs without a version number will be interpreted according to these draft specifications. This should assure early adopters that the standardization process would not undermine their efforts.

In the long term, only an evolving OpenURL standard can be successful: it must continually adapt to new technologies. It is easy to get caught up in the minutiae of encoding issues. However, encoding is intimately tied to current technology and is, therefore, not the proper foundation for a long-term evolutionary process. For Version 1.0, the committee intends to put in place the theoretical and fundamental concepts that are independent of technology.

At the core, the fundamental issue is which metadata of which possible entities need to be described. In turn, this depends on what an OpenURL is supposed to be. The initial discussions led to the following definition of an OpenURL:

An OpenURL is a transportation mechanism for metadata that describe

- *one or more referents and*
- *zero or more other entities that define the context in which the reference to the referents occurs or in which the transportation of metadata takes place.*

This framed the discussion led to the following result:

In an OpenURL, we must be able to describe the following entities:

- *Referent*
- *Resolver*
- *Requester*
- *Referrer*
- *Referring-entity*
- *Service-type*

Each of these entities can potentially be described in several different ways. The fundamental metadata-description mechanisms (or descriptors) include:

- *Id*
- *Metadata-description (by value)*

- *Metadata-description-pointer (by reference)*
- *Private-zone*

For each entity type, a menu of appropriate descriptors will be available. For example, it is likely that the resolver must be described by means of the id descriptor. That would be overly restrictive for the referent, and it is likely the referent may be described by any of the four available descriptors. For details, please consult NISO committee AX documents [\[NISO Committee AX\]](#) and Van de Sompel and Beit-Arie's theoretical OpenURL framework [\[Van de Sompel and Beit-Arie 2001\]](#), which formed the basis of committee discussions.

Conclusion

OpenURL is the beginning of an evolution that will increase the power of web links. With OpenURL, web links

- Are context sensitive.
- Deliver narrowly targeted and appropriate services.
- Have a longer useful life.
- Provide connections to services and information that did not yet exist at the time of writing the documents.

Right now, OpenURL improves the functionality of bibliographies. In the future, it will improve the functionality of the complete ETD, because not only can one provide services for citations but for subject headings, chemical formulas, genomes, products, patents, etc.

OpenURL is only one of the fundamental reasons why the scholarly record should be preserved in well-constructed OpenURL-aware electronic documents. Because beginning researchers are not only open to use these new technologies, they are eager to use them, ETDs are the best place to start this (r)evolution.

Related Web Sites

Endeavor Information Systems, Inc. LinkFinderPlus.
<http://www.endinfosys.com/prods/lfwhatis.htm>

Ex Libris (USA), Inc. SFX. <http://www.sfxit.com/>

Ex Libris (USA), Inc. OpenURL JavaScript.
<http://demo.exlibrisgroup.com:8888/OpenURL/javascript.html>

Fretwell-Downing, Inc. Open Linking Technology.
<http://www.fdusa.com/products/olt.html>

NISO. The Web site of NISO. <http://www.niso.org>

NISO Committee AX. The web site of NISO Committee AX on OpenURL Standardization. <http://library.caltech.edu/openurl>

<http://resolver.library.caltech.edu/caltechLIB:2002.002>


Openly Informatics, Inc. 1Cate, jake.openly.com, and link.openly.com.
<http://www.openly.com>


Van de Sompel, Herbert; Hochstenbach, Patrick and Beit-Arie, Oren. May 2000.
OpenURL syntax description. <http://www.sfxit.com/openurl/openurl.html> or
http://library.caltech.edu/openurl/Documents/OpenURL_Version_0.1.mht


Van de Sompel, Herbert and Hochstenbach, Patrick. 2000. Cookiepusher document.
<http://www.sfxit.com/openurl/cookiepusher.html>


Bibliography

Powell, Andy. OpenResolver: A simple OpenURL Resolver. [Ariadne](http://www.ariadne.ac.uk/issue28/resolver/intro.html). 22-June-2001.
Issue 28. <http://www.ariadne.ac.uk/issue28/resolver/intro.html> 

Van de Sompel, Herbert and Hochstenbach, Patrick. 1999a. Reference Linking in a
Hybrid Library Environment. Part 1: Frameworks for Linking. [D-Lib Magazine](http://www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt1.html). 5(4).
http://www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt1.html 

Van de Sompel, Herbert and Hochstenbach, Patrick. 1999b. Reference Linking in a
Hybrid Library Environment. Part 2:SFX, a Generic Linking Solution. [D-Lib Magazine](http://www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt2.html).
5(4). http://www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt2.html


Van de Sompel, Herbert and Hochstenbach, Patrick. 1999c. Reference Linking in a
Hybrid Library Environment. Part 3: Generalizing the SFX solution in the "SFX@Ghent
& SFX@LANL" experiment. [D-Lib Magazine](http://www.dlib.org/dlib/october99/van_de_sompel/10van_de_sompel.html). 5(10).
http://www.dlib.org/dlib/october99/van_de_sompel/10van_de_sompel.html 

Van de Sompel, Herbert and Beit-Arie, Oren. 2001. Generalizing the OpenURL
Framework beyond References to Scholarly Works, The Bison-Futé Model. [D-Lib
Magazine](http://www.dlib.org/dlib/july01/vandesompel/07vandesompel.html). 7(7).
<http://www.dlib.org/dlib/july01/vandesompel/07vandesompel.html> 

Van de Velde, Eric F. and Coles, Betsy. OpenURL-Aware ETDs. [Caltech Library System
Papers and Publications](http://resolver.library.caltech.edu/caltechLIB:2002.002). January 2002.
<http://resolver.library.caltech.edu/caltechLIB:2002.002>