SFX and the OpenURL Framework for the Hybrid Library

ELAG Conference
Prague, June 6-8, 2001

Tamar Sadeh
Ex Libris Ltd.

Contents

Figures

SFX and the OpenURL Framework for the Hybrid Library

Abstract

The concept of SFX is explained in the context of the OpenURL framework, the OpenURL standard, and the service components, such as SFX Server, that interact with OpenURLs. The structure and use of an OpenURL are described, as well as the compatibility of the OpenURL framework with other linking solutions.

What Is SFX?

SFX is about linking. Context-sensitive reference linking. The term "SFX" is often bandied about these days, and this article attempts to explain what it means. Selected by a team led by Dr. Herbert Van de Sompel at the University of Ghent, "SFX" stands for "special effects." The team aimed to create the same magic in the electronic information environment that special effects produce in the movie industry.

As a prelude to finding out what SFX is all about, let's look at the following case study.

As a psychiatry researcher, I am currently investigating the relationship between the use of the drug venlafaxine and the occurrence of unusually vivid dreams. Not particularly technically oriented, I often find myself intimidated by the challenges that I face in trying to obtain the services I need—namely, connecting to different electronic resources; remembering user names, passwords, and syntax; copying and pasting information from one system to another; and even repeating the same procedure in different systems. For example, the citation from PubMed shown below offers me some useful information.

> Psychol Rep 2001 Apr;88(2):514-6
>
> **Risk-taking, death anxiety, and dreaming.**
>
> Kroth J, McDavit J, Brendlen C, Patel A, Zwiener L.
>
> Graduate Division of Counseling Psychology and Education, Bannan 222, Santa Clara University, Santa Clara, CA 95014, USA.
>
> PMID: 11351900

Although in this case PubMed does not furnish a link to the full text of the article nor any further information specifically about the article, it does provide the abstract, which is a good starting point. After reading the abstract, I conclude that the article will be helpful.

Before going any further, I would really like some additional information about the article and its authors. The best source is the ISI® Web of Science®, where I

can find out whether the article has been cited, whether the authors have published other articles on similar subjects, and whether this field is their primary research area. So now I need to log in to the Web of Science, enter my user name and password, and start searching for the information. Not as easy as it sounds! I copy and paste the title as is. No results. I go back to PubMed and copy the name of the first author. No results. I try the second author, and then the third one. Nothing! Four rounds of copying and pasting have not yielded any promising results.

If I want to read the article, I must write down (or copy to the clipboard) the citation information and then log in again to my library's PAC. There I will have to type in the information needed to conduct a search for the printed item or its electronic version. Or, if all else fails, I will invoke the institution's document-delivery service form and type the information (or copy and paste it) once again.

Since I'm eager to find out more about the study, I decide to write the first author and ask for some information. How do I find the author's e-mail address? Perhaps I can use a Web service, where I will once more paste the author's name and try my luck. It also occurs to me that one of the authors might have written a book on the subject. Maybe I should look in my library's PAC or a union catalog, or even an Internet bookstore. And when I've completed my research, perhaps I can send my manuscript to the same publisher. So how do I contact the publisher? There is probably a site where I'll be able to find the necessary information.

These are just some of the issues that a researcher deals with when attempting to find information and obtain services in the vast resources available today through the Internet. SFX assists such researchers by integrating different resources and services into what appears to users as one environment.

The Current Situation: Closed, Non-Context-Sensitive Linking Frameworks

Many information providers today offer reference-linking services as part of their Web-based services. Most of these linking services typically consist of a link from metadata to the full content described by the metadata. That is, the provider of the metadata to the end user is also the provider of the reference-linking services, which address only a small portion of the linking possibilities available via the Internet.

Such frameworks have two major drawbacks:

- The links are typically not context sensitive; that is, all users see the same links, no matter what their affiliation. In addition, the links are limited in scope. Often, they lead the user to resources of the same information provider; at times, they may link the user to sites that are selected on the basis of business agreements between information providers.
- Users are not offered additional services, referred to by the team that developed SFX as "extended services," which can be of various types. To name just a few examples, these services could enable users to search in citation databases such as the Web of Science or the Stanford Public Information Retrieval System (SPIRES) at the Stanford Linear Accelerator Center (SLAC) (http://www.slac.stanford.edu/spires/hep/) for information about authors and articles; to locate the e-mail address of an author; to find holdings of a journal or a book in various resources such as the user's

library PAC or a union catalog; to use a Web search engine to locate more information regarding a specific subject; and to find information about a patent or a drug that was cited.

The links in the currently available frameworks are "closed"; that is, the end user's institution has either no control or only limited control over the links and the targets they lead to. Furthermore, the institution is unable to add other links that may be of interest to its users.

A special issue that arises from closed, non-context-sensitive frameworks is the "appropriate copy" problem. Information providers typically offer links to full text in the publisher's repository. However, as the scholarly information environment is mostly a licensed one, the provision of links that best serve the policy of the user's institution and best utilize its subscriptions assumes great importance. To make the most of the price they pay for licensed scholarly materials, institutions obtain them from various providers, including both primary and intermediate providers, and in either electronic or printed form; and the materials may reside either in remote resources or on the institution's servers. For instance, an institution may choose to acquire a Wiley journal directly from Wiley or via Swetsnet*Navigator*, or may decide to furnish its users with only the printed version. Another example is the subscription to the Web of Science, which can be accessed from either an ISI server or an on-site installation. Users should be directed to the appropriate Web address for obtaining the service they need or to the institution's document-delivery service, if the institution is not authorized to access the electronic form, if the institution wishes to handle the delivery by other means, or if no electronic version exists. Directing the user to the default location, typically the publisher's site, may result in the denial of the service or a request for payment even when the user is legitimately entitled to obtain the copy of the article.

The "appropriate copy" issue extends also to other types of linking services offered to users. For example, a researcher in the United States who requests a book review might be directed to amazon.com, whereas colleagues in the United Kingdom would be directed to amazon.co.uk. Similarly, an international company may work with several document-delivery services and on a case-by-case basis choose the most suitable one according to the user's geographic location.

The OpenURL Framework: An Open, Context-Sensitive Linking Framework

From 1998 to 2000, a team of researchers led by Dr. Herbert Van de Sompel at the University of Ghent in Belgium investigated the limitations of the existing linking framework in the scholarly information arena and suggested a new framework that addresses the problems they found. As a result of this project, the OpenURL framework and SFX technology were born. Linking server software called SFX was developed to realize the potential of the proposed framework. In February 2000, Ex Libris acquired the sole rights to the SFX reference-linking software solution from the University of Ghent.

The OpenURL framework is based on several assumptions:

1. The provision of services can be detached from the description of the work; similarly, the provider of the citation information (the metadata record, reference, and so forth) need not be the provider of the services.

2. The metadata is the key for the generation of services; and since the metadata is in the hands of the information provider, it is that information provider who can best furnish the relevant metadata.
3. Localized, appropriate services are best defined by institutions on the basis of their policies and subscriptions.
4. A layer of such localized services can be created in addition to the layer of default services offered by information providers.

As suggested by the OpenURL framework, reference linking based on specific metadata in the scholarly environment should not be considered part of that metadata but rather a service built on the metadata. Consequently, the provider of such services can be any third party, and the way in which they are provided is through a service component, or "link server," that creates an overlying level of services along with the services offered by the information provider.

The service component is a software component that, on the one hand, receives the metadata for which services are required (the metadata coming from the information provider) and, on the other, is sensitive to the user's context—that is, applies user-specific information, such as the policies and subscriptions of the user's institution, when generating services. Within the OpenURL framework, an OpenURL is the standard manner of delivering metadata from an information provider to a service component. It is up to the service component to obtain the information regarding the user's affiliation; the OpenURL framework does not specify standards for this function.

Figure 1 illustrates the role of the service component in the OpenURL framework. Whereas an information provider supplies default links from a source (the initiator of services) to a number of targets (resources to which a service takes a user), the same information provider may deliver the metadata in the form of an OpenURL to the relevant service component. The service component, in turn, generates context-sensitive reference linking based on its knowledge of the user. These services can link the user to other targets, as appropriate, within a larger range of resources available to that specific user.

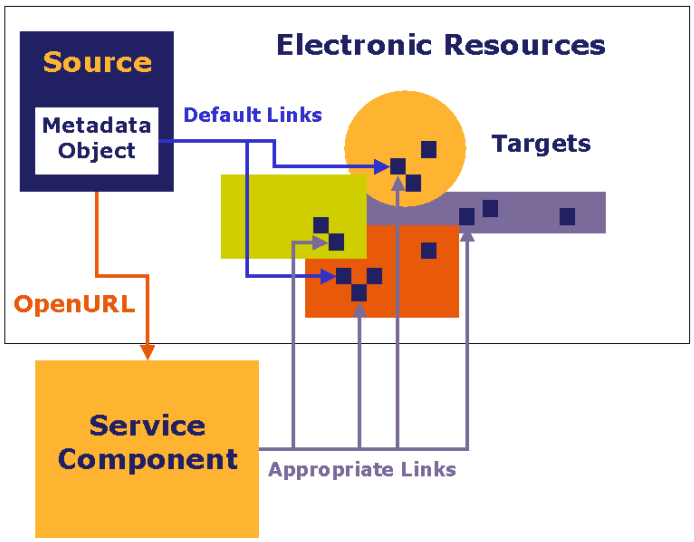

Figure 1. Role of the service component

The OpenURL framework can support a range of different service components. The one to which this article refers is SFX Server, already operational at a number of sites around the world. Other OpenURL-compliant service components are now emerging as well.

Information providers who wish to participate in the OpenURL framework must be capable of generating an OpenURL that will be directed to the service component available to the user who has requested the services. For service components to be compatible with the OpenURL framework, they must be able to work with any content and services. The service components are not to be restricted to content and services offered by specific information providers.

To expose their metadata, information providers add a hook to their system for each metadata object. Such a hook is typically represented in the system's interface by a button. By clicking the button, the user requests extended services for the specific metadata object. As a result, the information resource delivers the OpenURL to the service component, which parses the metadata, evaluates it, and generates the appropriate services.

The OpenURL syntax description specifies the metadata elements (for example, ISBN, ISSN, title, and subject) that are relevant to the successful generation of extended services. In any given OpenURL, the elements will depend on the nature of the metadata object. For instance, metadata describing a book will include its ISBN, whereas metadata describing an article will include its ISSN, volume, issue, starting page, and so forth. The OpenURL itself may be considered "meta-metadata," according to which specific elements of metadata are selected and transported in a standard format that will enable services to be generated afterwards.

We may note that the quality of the metadata plays a major role in the ability of the service component to provide the extended service. The more detailed the metadata is, the richer the linking services. For example, if only the name of a journal is provided (or the ISSN, for that matter), a link to the home page of the journal may be furnished. However, if the volume, issue, and page number are given as well, a link to the full text of the article can also be offered (depending on other factors, such as a valid subscription to the journal and a proper link-to syntax offered by the target of the link).

Using a pragmatic approach to the selection of metadata elements from the list of those constituting a bibliographic record, the OpenURL standard specifies only a minimal set as those required for generating common services. Since this set might not satisfy all the requirements for generating services, the service component may have to fetch more metadata to provide a certain service. Thus, the OpenURL syntax also provides for the inclusion of a unique identifier for the metadata object. For instance, the OpenURL syntax supports the delivery of metadata elements related to only one author. If the service component is configured to generate services for additional authors, it must first obtain the relevant metadata object.

Note that the OpenURL provides for outbound linking, that is, linking from an information resource (the source) to the service component. Inbound linking—from the service component to the various targets that provide the actual services (such as publishers' repositories of articles, Web search engines, citation databases, library catalogs, and Internet bookstores)—is configured according to

the specific syntax provided by the targets. The targets need not comply with any standard format, although standards in this area would be welcome.

The OpenURL Standard

The OpenURL standard provides a format for transporting metadata about objects between information services. It is now on its way to becoming a NISO (National Information Standards Organization)-accredited standard (see www.niso.org/commitax.html).

What Is an OpenURL?

An OpenURL permits the delivery of metadata from an information resource to a service component. Two basic elements make up an OpenURL: (1) the base URL, which is the Web address of the relevant service component; and (2) the content. The content could be the metadata of the specific object or reference to the metadata object. Since the metadata resides within the information provider's system, the information provider is the one who should generate the OpenURL, in the standard format. In this manner, the service component is spared the need to know the source of the information. Thus the advantage of the system's openness: any resource that implements an OpenURL can send its metadata to a service component, which will always handle the metadata in the same, standardized way.

Metadata is delivered via an OpenURL either by "value," that is, through the explicit inclusion of the metadata elements (such as the ISSN or ISBN, author name, and title) in the URL, or by "reference"—through the inclusion of a unique metadata object identifier that the service component can use to extract the metadata as necessary. In essence, an OpenURL can include both the identifier and the actual metadata. The identifier may point to the original resource or to another one in which the metadata is available, such as the CrossRef database.

For a detailed description of the OpenURL syntax, see http://sfx1.exlibris-usa.com/OpenURL/OpenURL.html.

Examples

The following examples illustrate several ways of using the standard syntax to build OpenURLs:

1. An OpenURL that could be generated for the article cited earlier, in the introductory section ("Risk-taking, death anxiety, and dreaming"):

   http://sfx.anywhere.edu/library?genre=article&atitle=Risk-taking%2C%20death%20anxiety%2C%20and%20dreaming%2E&stitle=Psychol%20Rep&date=2000-04&volume=88&issue=2&spage=514&epage=516&aulast=Kroth&aufirst=J

   In this case, the base URL—the address of the service component available to the user who requested the service—is http://sfx.anywhere.edu/library?

   Note that it is up to the information provider to create the OpenURL; therefore, other representations of the specific metadata object might exist. For instance, the OpenURL shown in example 2 is what Ex Libris's MetaLib information portal generates for the same metadata object.

2. An OpenURL generated for a record whose origin is SilverPlatter's Medline:

   http://sfx.anywhere.edu/library?sid=metalib:SILVER_MEDS&genre=&isbn=&issn=0033-2941&date=2001&volume=88&issue=2&spage=514&aulast=kroth&aufirst=%2DJ&auinit=&title=Psychological%2Dreports&atitle==Risk-taking%2C%20death%20anxiety%2C%20and%20dreaming.&pid=DocNumber=00024137,lp=www.metalib.com,Port=9919,ServiceClass=ALL

   In addition to the metadata tags, this URL includes the source identifier (`sid=metalib:SILVER_MEDS`) and the private identifier of the specific record (`pid=DocNumber=00024137,lp=www.metalib.com,Port=9919,ServiceClass=ALL` for the first record). A combination of both the metadata tags and the identifiers in the OpenURL enables the service component to display certain services based on the metadata elements that have been parsed from the OpenURL (in this example, since we have the ISSN, volume, issue, and starting page, a link to the full text of the article can be generated). Other services, such as those based on names of other authors, are displayed only after more metadata elements have been fetched by means of the identifier. Note that the private identifier, as demonstrated in this example, can be complex and can include any information that the source needs for locating its record. This type of usage implies that the service component has a parser that can extract the metadata from the source; we are thus looking at a proprietary solution rather than the generic one that is encouraged by the theory of the OpenURL standard.

3. The use of another type of identifier, namely a digital object identifier (DOI). The OpenURL for the citation "The DOI Handbook, by Dr. Norman Paskin, Version 1.0.0, February 2001, International DOI Foundation (IDF) DOI:10.1000/182" would be:

   http://sfx.anywhere.edu/library?id=doi:10.1000/186

   Note that in this case, no metadata tags were delivered via the OpenURL. However, they can be obtained from the CrossRef database, where they are stored in a known format.

The Role of the Service Component

After receiving an OpenURL as input, the service component gathers the metadata either by parsing it from the OpenURL or by fetching it via the identifier supplied by the OpenURL, or both. At this stage, the service component can analyze the metadata, determine the services that can be generated, and generate them on the basis of the institution's strategies and subscription policies. Typically installed at the user's institution, the service component needs to know the information required for the successful generation of the services—the methods of linking to all the targets and every detail concerning the institution's subscriptions and policies. The goal is to provide the users with only the services that are actionable and thus save them the frustration of being denied access to a service. For instance, if an institution began subscribing to a certain electronic journal in 1998, a link to the full text of articles should be provided only for articles published from 1998 on.

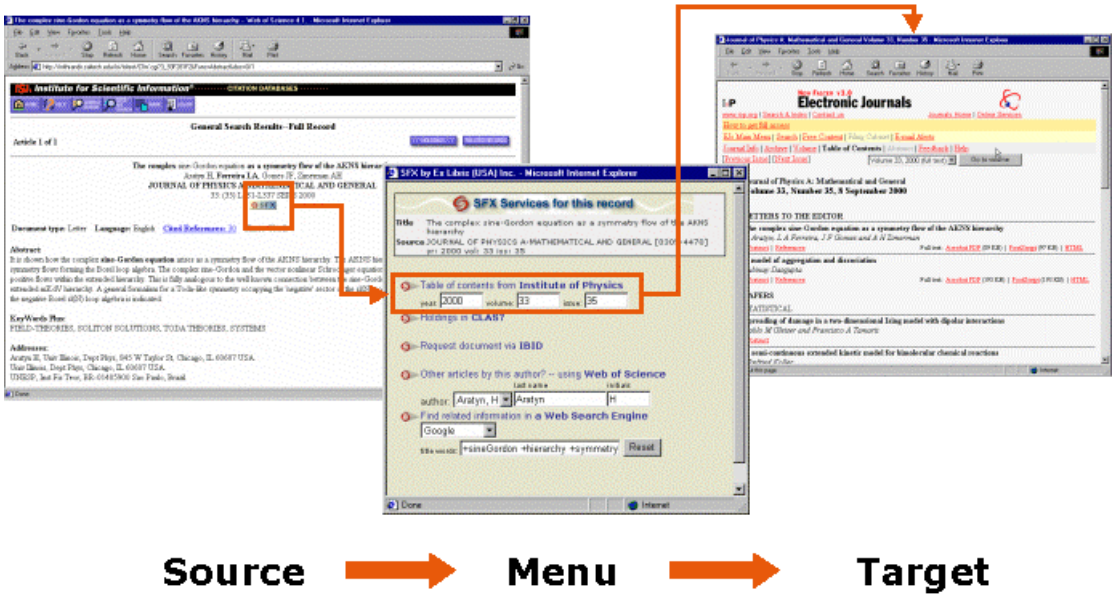

Figure 2. User's interaction with the service component

SFX Server

SFX Server is the Ex Libris manifestation of a service component. It expects its input in the form of an OpenURL consisting of metadata objects and identifiers. The metadata is then parsed from the OpenURL, fetched from the original resource, or fetched from another resource by means of the identifier.

Once the metadata is available to SFX Server, the server analyzes it and uses its knowledge of the institution's policies and subscriptions to dynamically generate a set of extended services appropriate to the specific metadata and the specific user. For instance, if the metadata object is an article in a journal to which the user's institution has a subscription and which is published in both electronic and print forms, the services for this particular user may include a link to the appropriate copy of the full text of the article and to the relevant holdings information in the library's PAC. If, on the other hand, the article was not published electronically or the library does not have a subscription to its electronic version, the library can opt for offering a document-delivery service to the user in addition to providing the information on holdings. Note that a document-delivery service might be realized as a document-delivery form that already contains the details of the metadata object relevant to the order.

Clearly, many other services can be offered to any type of Web resource that provides for link-to or search syntax. Figure 3 shows an example of SFX services on a screen from the University of Ghent. To see the types of services the University of Ghent offers for different types of metadata objects (such as books and journals), look at the university's PAC at http://www.lib.rug.ac.be:4505/ALEPH/-/start/rug01eng.

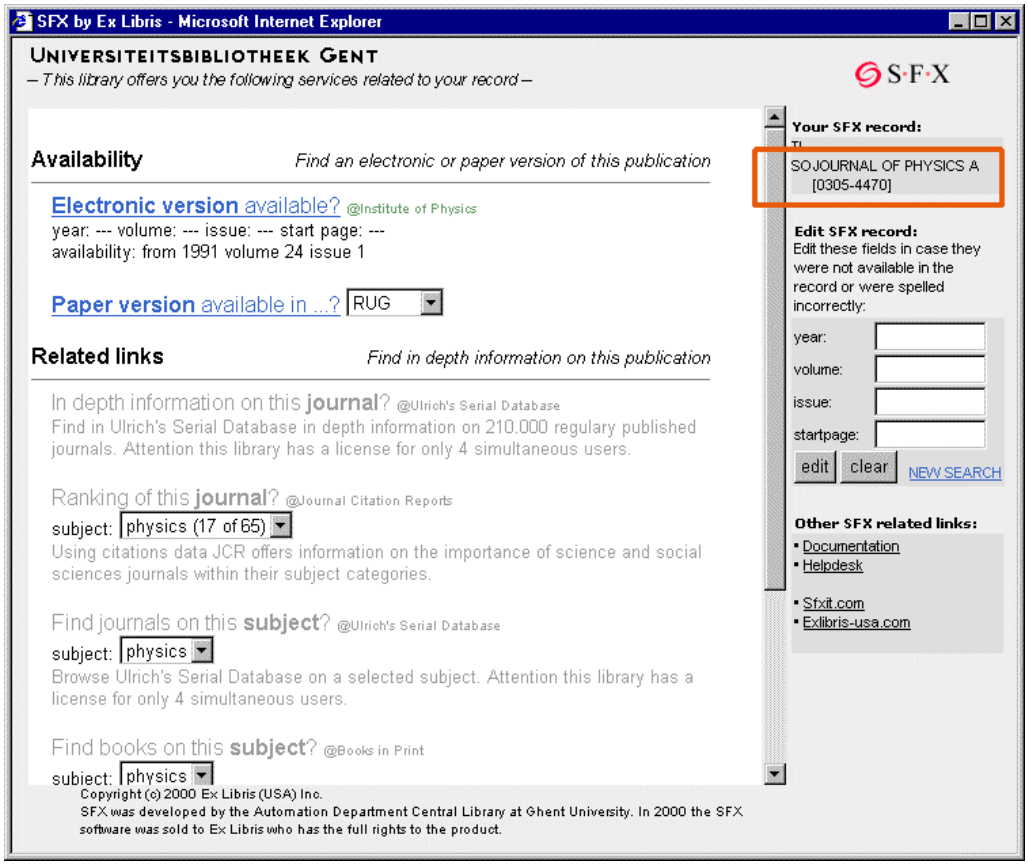

Figure 3. Reference-linking service generated by SFX Server at the University of Ghent for Journal of Physics. A: Mathematical and General

It is important to note that the extended services that are offered depend on institutional decisions and do not require any restrictions on the linking syntax to the various targets of the links. Practically any Web resource and service can be configured, with the "depth" of the link varying according to the quality of the metadata available to the service component and the level permitted by the target.

Let's take a look at an example of a citation and its links. The citation is
Proc. Natl. Acad. Sci. USA, Vol. 98, Issue 10, 5419-5425, May 8, 2001
The future of coral reefs
Knowlton N

The links provided for this citation are shown below. The bold font indicates the parameters that were embedded in the formula according to which the link was computed.

- Link to the full text:
  http://www.pnas.org/cgi/content/full/**98/10/5419**
- Link to the abstract:
  http://www.pnas.org/cgi/content/abstract/**98/10/5419**
- Link to the holdings information at the Ghent ALEPH PAC:
  http://www.lib.rug.ac.be:4505/ALEPH/-/start-ext/rug01?find-a?F1=wis&V1=**0027-8424**

- Link to a Google search:
  http://www.google.com/search?q=**The%20future%20of%coral%20reefs**

- Link to the SLAC/SPIRES HEP citation database:
  http://www.slac.stanford.edu/spires/find/hep/www?AUTHOR=Knowlton

- Link to PubList:
  http://www.publist.com/search/search.asp?ISSN=**00278424**&SearchType=Adv

The OpenURL Framework and the DOI/CrossRef Linking System

As an open system, the OpenURL framework is compatible with other solutions for reference linking, such as the DOI/CrossRef linking system. Currently, more than 70 leading publishers, accounting for over 3,800 journals, participate in CrossRef. For each published work, a participating publisher receives a unique identifier—a DOI—that is assigned when the publisher submits the metadata to CrossRef. This metadata, along with the DOI and the URL supplied by the publisher, is then stored in the CrossRef database, which holds over three million records for articles at this time.

Publishers that are members of CrossRef may use the DOI as part of a publication's citation information, typically inserted in references in the bibliographies of works they publish. To add the DOI to the citation information, the publisher sends the metadata elements from the citation to the CrossRef database ([1] in Figure 4) and as a result receives the DOI of the record ([2]). This DOI is then embedded within the publisher's user interface. Figure 5 shows an example of a citation that includes such information.

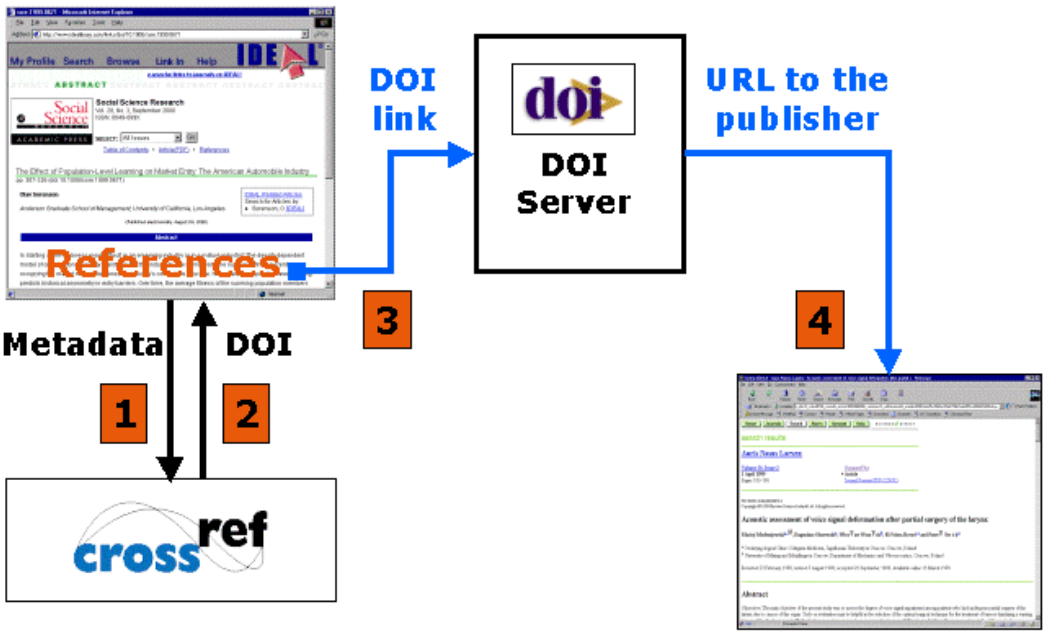

Figure 4. DOI/CrossRef framework

Once the user clicks the DOI link ([3] in Figure 4 above), the DOI is sent to the DOI server, which resolves the DOI to the URL that was furnished by the original publisher and then transfers the user to that location ([4]).

Aratyn H 1999 *J. Geom. Phys.* **30** 295 (MathSciNet Review) (CrossRef Link)

Figure 5. Citation from reference section of an article published by the Institute of Physics

The limitations of this solution take us back to our earlier discussion of the drawbacks of closed, non-context-sensitive linking frameworks. First, the location of the full text of the article is determined by the primary publisher and is stored as a static link (within the DOI Handle System). Hence, if the institution subscribes to an alternative provider of the article's full text or has an on-site copy of the article, the link will yield a rejection of the service even if the user is entitled to obtain the full text. A similar problem can occur when the library does not have a license to any electronic version of the article and wishes to direct the users to its printed version or its document-delivery service; the DOI framework does not allow the library to do this. Finally, no services other than the full text are currently available in the DOI/CrossRef framework.

Use of an OpenURL can overcome this problem. In an important prototype (see http://www.dlib.org/dlib/may01/brand/05brand.html), the DOI proxy server was rendered OpenURL-aware. When a user clicks the DOI link ([1] in Figure 6), the DOI is sent to the DOI proxy server as usual. Because the DOI proxy server is OpenURL-aware, it may now identify users as coming from an institution that has a service component. In such cases, the DOI proxy server hands the request over to the user's service component ([2] in Figure 6) by transmitting an OpenURL that includes the DOI as a parameter. Once it has the DOI, the service component can obtain the metadata from the CrossRef database ([3] and [4]) and generate the services that are appropriate for the user ([5]).

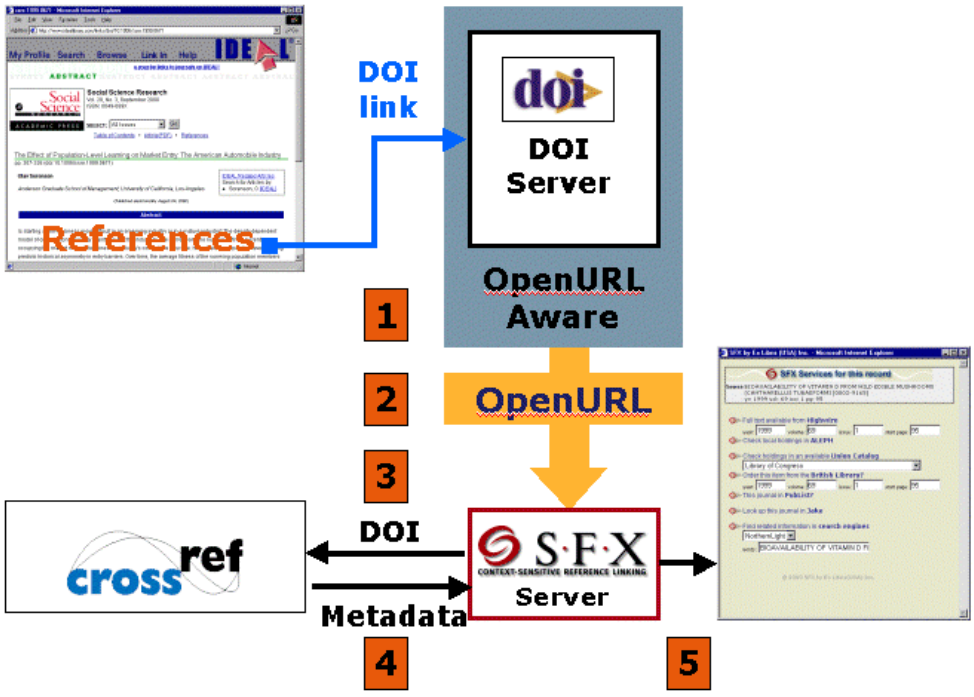

Figure 6. DOI/CrossRef integration with the OpenURL framework

If the user is not identified as having access to a service component, the DOI proxy server uses its default resolution process, via the DOI Handle server, which yields the URL that the publisher initially submitted to the CrossRef database.

Conclusions

The OpenURL framework as described here offers institutions the ability to define and control the reference-linking services they would like to grant their users. This level of localized services is provided in addition to the services that information providers offer and spares the information providers the need to accommodate the increasing demand for localized, extended services.

For users, this framework provides a standardized interface to extended services, whether the users are within their local environment, such as their library's PAC, or exploring other, remotely held, resources that are participants in this framework. The service component (link server) enables users to obtain the services that are the most relevant to them—and all integrated in the same environment.

As seen here, the OpenURL framework is compatible with other linking solutions. A good example is the DOI/CrossRef framework; the OpenURL framework actually complements it by permitting the generation of context-sensitive extended services within that framework.

The success of the OpenURL framework depends on the cooperation of information providers. Many of the key information providers have already implemented the OpenURL or committed to do so.

SFX Server is setting the standards for OpenURL service components, and many installations of SFX Server are already operational in institutions around the world.

References

Brand, Amy. 2001. CrossRef turns one. *D-Lib Magazine* 7, 5. [http://www.dlib.org/dlib/may01/brand/05brand.html]

Caplan, Priscilla, and William Y. Arms. 1999. Reference linking for journal articles. *D-Lib Magazine* 5, 7/8. [www.dlib.org/dlib/july99/caplan/07caplan.html]

OpenURL Generator
[http://sfx1.exlibris-usa.com/OpenURL]

OpenURL-aware resources.
[http://sfx1.exlibris-usa.com/sources-list.html]

Van de Sompel, Herbert. July 2000. CrossRef -- DOI -- OpenURL -- SFX demo using IDEAL metadatabase & enhanced DOI proxy.
[http://www.sfxit.com/crossref/prototype1.html]

Van de Sompel, Herbert, and Oren Beit-Arie. 2001. Open linking in the scholarly information environment using the OpenURL framework. *D-Lib Magazine* 7, 3.
[http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html]

Van de Sompel, Herbert, Patrick Hochstenbach, and Oren Beit-Arie. 2000. OpenURL syntax description.
[http://www.sfxit.com/openurl/openurl.html]

Van de Sompel, Herbert, and Patrick Hochstenbach. 1999a. Reference linking in a hybrid library environment. Part 1: Frameworks for linking. *D-Lib Magazine* 5, 4.
[www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt1.html
Van de Sompel, Herbert, and Patrick Hochstenbach. 1999b. Reference linking in a hybrid library environment. Part 2: SFX, a generic linking solution. *D-Lib Magazine* 5, 4.
[www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt2.html]

Van de Sompel, Herbert, and Patrick Hochstenbach. 1999c. Reference linking in a hybrid library environment. Part 3: Generalizing the SFX solution in the "SFX@Ghent & SFX@LANL" experiment. *D-Lib Magazine* 5, 10. [www.dlib.org/dlib/october99/van_de_sompel/10van_de_sompel.html]