

Two Clicks Max:

dissecting link resolvers
&
considering their user benefits



a presentation for K-State Libraries by
Jamene Brooks-Kieffer, candidate
Resource Linking Librarian
July 10, 2006



Agenda

Part One:

- Link resolver types
- Link resolver components
- Internal workflow
- Demonstration

Part Two:

- Who are users?
- Benefits & Drawbacks
- Conclusions
- Questions

So many choices...



- What kind of link resolver?
 - Proprietary
 - commercially available, fee-based
 - Open source
 - varies – free, fee based, or built in-house

ARTICLE
LINKER

1cate

SFX

LS

OL2

V-link

Open
Resolver



ArticleLinker – from SerialsSolutions

1Cate – from Openly Informatics, now owned by OCLC

SFX – from ExLibris

LS = LinkSource – from EBSCO

OL2 – from Fretwell Downing; add-on to ZPortal federated search product

V-link – from Geac Library Solutions

OpenResolver – open-source demo from UKOLN (UK Office for Library Networking)
@ University of Bath

GODOT/CUFTS – open-source from Simon Fraser University Library

Singer, R. (2006, February). Helping you buy: link resolver tools. *Computers in Libraries*, 26, 2, pp. 15-23. Retrieved June 22, 2006 from Academic Search Premier.

Lewis & Clark College Watzek Library built their own open-source resolver, but it has no logo.

Dahl, M. (2004, February). Building an OpenURL resolver in your own workshop. *Computers in Libraries*, 24(2), pp. 6-11. Retrieved June 23, 2006 from Academic Search Premier.

Proprietary vs. Open-source



- Three components for most:
 - Participation in OpenURL standard (NISO Z39.88)
 - data transport rules
 - Knowledgebase
 - brains
 - Search script
 - muscle

Some resolvers do not rely on a knowledgebase, but this is rare (Singer, 2006, p.22). Almost all rely on some ratio of these three components.

Singer, R. (2006, February). Helping you buy: link resolver tools. *Computers in Libraries*, 26(2), pp.15-23. Retrieved June 22, 2006 from Academic Search Premier.

OpenURL Standard



- OpenURL 0.1 - first version
 - Easy, but not extensible beyond serials data
 - `http://www.library.org/resolver.cgi?title=Computers+In+Libraries&issn=1041-7915&date=2004&volume=24&issue=9&aulast=Ferguson&atitle=October:+OpenURL+Link+Resolvers&spage=17`

(Singer, 2006, p.15)

Singer, R. (2006, February). Helping you buy: link resolver tools. *Computers in Libraries*, 26(2), pp.15-23. Retrieved June 22, 2006 from Academic Search Premier.

Walk through URL elements, but don't linger on any one thing.

Refer to database model: entity & attributes, if that helps anyone;

Or compare to ILS modules that describe a patron or book or course in terms of specified categories.

OpenURL Standard



- OpenURL 1.0, aka NISO Z39.88-2004
 - Approved April 2005
 - Extends beyond serials data
 - http://www.library.org/resolver.cgi?ctx_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:journal&rft.jtitle=Computers+in+Libraries&rft.issn=1041-7915&rft.date=2004&rft.volume=24&rft.issue=9&rft.aulast=Ferguson&rft.atitle=October:+OpenURL+Link+Resolvers&rft.spage=17
(Singer, 2006, p.15), (NISO, 2005, p.13)

Singer, R. (2006, February). Helping you buy: link resolver tools. *Computers in Libraries*, 26(2), pp.15-23. Retrieved June 22, 2006 from Academic Search Premier.

NISO: National Information Standards Organization. (2005). The OpenURL Framework for context-sensitive services: an American national standard: ANSI/NISO Z39.88-2004. Bethesda: NISO Press. Retrieved June 29, 2006 from <http://www.niso.org/standards/index.html>

rft – the article (Referent)

rft_val_fmt – specifies what parts of the standard this URL will be adhering to

Each attribute of the article (entity) is defined by the prefix rft. Other choices, depending on the purpose of the URL, would be:

req – Requestor – the final person destination – for an email address, for instance

res – Resolver – the linking server that handles the traffic on a particular Referent (entity).

(NISO, 2005, p.13)

Resolver Knowledgebase



- Knowledgebase = Database = Brains
- Global and local holdings and access data
- Depth/breadth determines worth to users
- Maintenance is value-added service
 - Proprietary LR vendors often manage global data
 - Purchase vendor/publisher relationships, data expertise

Global data: every instance of available data located in every resource held in the KB

Local data: the Library's instances of available data located in their versions of the resources held in the KB.

Ex:

Academic Search Premier has years 1960-2002 of Nature available. (Global)

Because of the Library's purchase agreements, the Library only has access to years 1995-2002 of Nature in Academic Search Premier. (Local)

Local data also encompasses permissions information – that keeps the resolver returning only allowed results.

Access rules such as only allowing Engineering students to access the IEEE database are included here.

Vendor management of the knowledgebase is often one good reason why libraries choose to purchase or contract with a proprietary link resolver company.

Much time and expertise is involved in keeping both global and local holdings updated in even a small knowledgebase.

Resolver Search Script



- Between Web pages and Knowledgebase
 - PHP, PERL, or similar
- Manages incoming, outgoing OpenURLs
 - activates KB searches for incoming
 - formats outgoing into correct format
- Formats results for user

(Dahl, 2004)

Dahl, M. (2004, February). Building an OpenURL resolver in your own workshop. *Computers in Libraries*, 24(2), pp. 6-11. Retrieved June 23, 2006 from Academic Search Premier.

Think of search script as the “middleman” – it’s critical for correctly passing data back and forth among information providers, the library, and the user. It passes notes, but it doesn’t write any of its own.

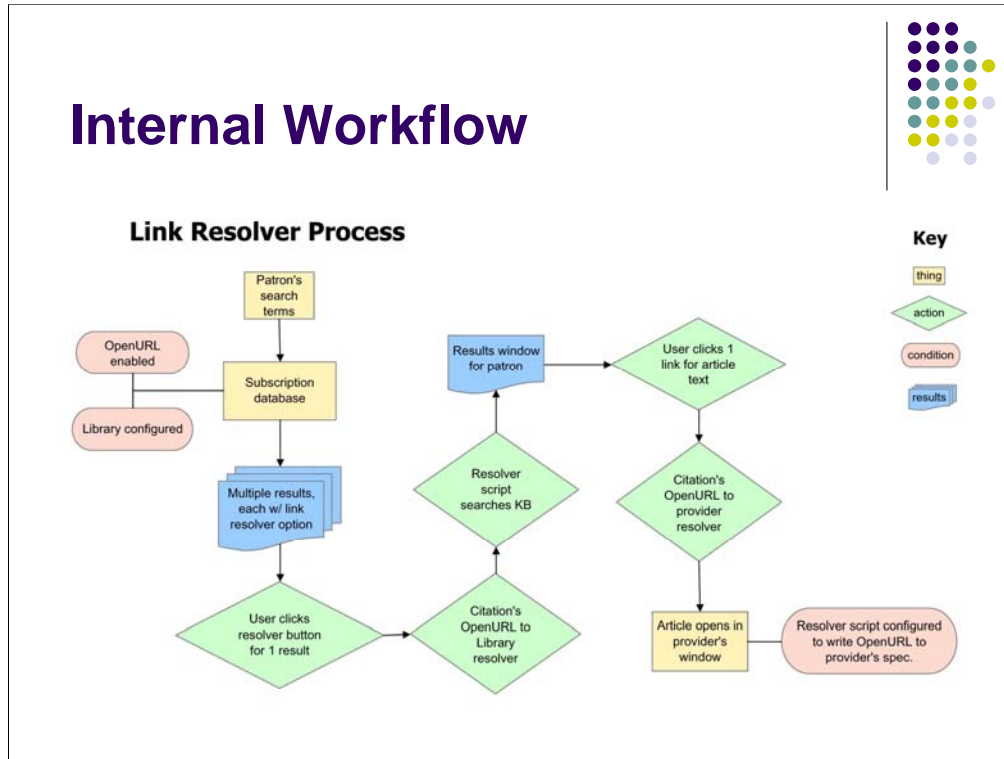
When OpenURLs come in from information providers, the script uses the data we saw encoded in the URL to initiate a search of the knowledgebase. The most likely open-source method is with SQL queries embedded into the script, but proprietary resolver scripts vary.

When OpenURLs go out from the library to information providers, the script writes the URL data into the format that the provider requires. This step is necessary to get the patron directly to the item-level of the provider’s site. Otherwise, they would be dumped onto the journal page or the provider’s home page.

When users are presented with lists of results from the knowledgebase, the script has formatted them in the manner customized by the library.

PHP or PERL are not the only languages you can write a resolver script in, but they are common for writing Web/database middleware – that’s all this script is, really. Many open-source resolvers use techniques and languages familiar from other aspects of Web development. Again, proprietary resolver scripts vary.

Internal Workflow



Point out path of data travel, with key points:

*Conditions must be in place for the process to work

*Diagram assumes subscription database, but search terms could be put into any information provider that is OpenURL enabled and correctly configured by the library

*The process takes place for every article or other item the patron wants to view via the link resolver button

*Two link resolvers are at work here – the Library's and the database vendor's (or other information provider's) – thus the difference between incoming and outgoing OpenURLs.

*If article-level linking fails, patron is probably still taken to vendor's page referencing the journal title. At the very least, the patron is taken to the vendor's database start page, where s/he will have to put in the citation manually.

Dahl, M. (2004, February). Building an OpenURL resolver in your own workshop. *Computers in Libraries*, 24(2), pp. 6-11. Retrieved June 23, 2006 from Academic Search Premier.

Demonstration



- OpenResolver
 - open-source demo from UKOLN
(UK Office for Library Networking)
University of Bath
- <http://www.ukoln.ac.uk/distributed-systems/openurl/>

(Powell, 2001)

Powell, A. (2001, June 22). OpenResolver: a simple OpenURL resolver. *Ariadne*, 28. Retrieved June 26, 2006 from <http://www.ariadne.ac.uk/issue28/resolver/intro.html>

Open URL in separate Web page.

Demo displays resolver-specific results windows after consulting UKOLN knowledgebase.

Gives an idea of what user sees on surface, while flowchart shows what is going on underneath.

Follows process all the way through sending OpenURL to vendor and getting (maybe) article-level results page.

What else can it do?



- Extended services:
 - Check OPAC print holdings
 - Enable citation linking
 - Complete ILL forms
 - Place items on reserve
 - Save to EndNote, RefWorks, etc
- Require commitments:
 - Time, expertise
 - Configuration of unrelated systems
 - Training, marketing
 - Extensible resolver product

Singer (2006) p.22

Singer, R. (2006, February). Helping you buy: link resolver tools. *Computers in Libraries*, 26(2), pp.15-23. Retrieved June 22, 2006 from Academic Search Premier.

Go over some common examples of extended services; stress these are not the only extensions. Explain a few things:

Citation linking means linking from a citation in one item to the cited item (possibly somewhere else). Web of Science, or Web of Knowledge, is a database that uses citation linking well and is a good example.

Many libraries never implement all these services because they don't have time or expertise to make all the separate systems work with the resolver.

If the library wants to include print journal holdings with other OPAC print holdings, it is best if those holdings are already in MARC21 format (MFHD). If they are not, a retrospective project is another big commitment in time and expertise.

Because the features of a resolver aren't obvious once it is implemented, the library needs to constantly train users and market the service in order to make offering extended services worthwhile.

A resolver that will grow as new services are innovated is the best choice when implementing from the ground up – ensures the library will be able to offer extended services as it is able.



User Benefits

- Who are the users?
 - External
 - students
 - faculty/staff/administration
 - alums
 - community
 - consortia partners
 - Internal
 - you!

External Users



- Benefits:
 - Two clicks to e-content
 - Automated push to allowed providers
 - Less time repeating searches
 - Extended services unify separate pathways
 - OPAC (print)
 - electronic providers
 - ILL, course reserves

*We've seen the process that enables "two clicks max", but also seen how many things have to go right for users to only need two clicks. The potential is there, but of course, it can take more.

*Because the library has configured the resolver with local holdings data and local access rules, the user is directed only to those sources that the institution subscribes to. There is less of a chance of the user thinking s/he must purchase access to something that the library holds under subscription with another service.

*Since the resolver script tracks down paths to items from the results window, the search terms or citations are less likely to need rekeying by the user. And if the resolver is offering extended services, the data fields it tracks via the OpenURL should be auto-completed if there is a form for the user to complete.

*Of course, the resolver can only unify the parts that the library has configured to work with it.

Internal Users



- Benefits:
 - Unified e-content usage statistics
 - Solve appropriate copy and appropriate path problems
 - Centralize e-holdings updates
 - Centralize all holdings display
 - Standardized interaction with vendors, other information providers

*Statistics come for everything the resolver has entered and can see. Anything outside the resolver is separate.

*Because of local holdings data and local access rules, resolver only points users to items coming from sources they have subscription access to – they won't be confused into paying for items they should have gotten for "free" or be pointed to sources they don't have access to at all.

*For sources in the resolver, global holdings are handled by the link resolver vendor (if it's not home-built); local e-holdings all go through the same place.

*All holdings will be displayed only if the library has configured the OPAC and resolver to display print items, including print serials data from MARC21 format.

Compliance with either OpenURL 0.1 or 1.0 ensures that the data being moved back and forth will be readable by all parties.

Any Drawbacks?



- Usability issues
- Setup and maintenance commitments
- Complex system
 - Users need continuous training
- Invisible - little direct interaction
 - easy to forget until something goes wrong

Singer (2006)

Singer, R. (2006, February). Helping you buy: link resolver tools. *Computers in Libraries*, 26(2), pp.15-23. Retrieved June 22, 2006 from Academic Search Premier.

*The window of multiple results has proved to be a usability problem for some patron populations (Singer, 2006, p.22). The initial link resolver button is also confusing if the user does not already know what the icon means.

*We have discussed some of the commitments of time and expertise in setting up and maintaining a link resolver. When contracting with a vendor, some, but not all, of that responsibility can be handed to the vendor. The library still has plenty of local data and access rules to handle, even with a vendor's system.

*We have seen how complex these systems are. This presentation has been a very simple explanation – only scratching the surface. Even for users, the process of a successful interaction will involve training and careful documentation.

*One of the most interesting drawbacks of these systems is that they are pretty invisible. Once implemented, they disappear into the library's familiar system of Web pages, external information providers, and OPAC. We have looked behind the scenes to see how they work, but that is not something you will see in daily work with a resolver, unless you are one of the people maintaining it. And patrons never see it – the communication between resolver and information provider disappears when everything is working properly.

Conclusions and Questions



- Most link resolvers function with the same basic components
- Workflow is complex; many failure points
- Professional maintenance and support can be worth the money
- Users are everywhere – don't forget yourselves
- Don't let drawbacks overwhelm benefits

Sources Cited



Dahl, M. (2004, February). Building an OpenURL resolver in your own workshop. *Computers in Libraries*, 24(2) pp. 6-11. Retrieved June 23, 2006, from Academic Search Premier.

NISO: National Information Standards Organization. (2005). The OpenURL Framework for context-sensitive services: an American national standard: ANSI/NISO Z39.88-2004. Bethesda: NISO Press. Retrieved June 29, 2006, from <http://www.niso.org/standards/index.html>

Powell, A. (2001, June 22). OpenResolver: a simple OpenURL resolver. *Ariadne*, 28. Retrieved June 26, 2006, from <http://www.ariadne.ac.uk/issue28/resolver/intro.html>

Singer, R. (2006, February). Helping you buy: link resolver tools. *Computers in Libraries*, 26(2), pp. 15-23. Retrieved June 22, 2006, from Academic Search Premier.

Sources Consulted



American Library Association. (2006, Jan-Feb). Introduction. *Library Technology Reports*, 42(1), pp.5-7. Retrieved June 23, 2006, from InfoTrac OneFile.

American Library Association. (2006, Jan-Feb). Chapter 2: On the road to the OpenURL. *Library Technology Reports*, 42(1), pp.8-13. Retrieved June 23, 2006, from InfoTrac OneFile.

American Library Association. (2006, Jan-Feb). Chapter 3: The development of context-sensitive linking. *Library Technology Reports*, 42(1), pp.14-23. Retrieved June 23, 2006, from InfoTrac OneFile.

American Library Association. (2006, Jan-Feb). Chapter 4: Link-resolver products. *Library Technology Reports*, 42(1), pp.24-30. Retrieved June 23, 2006, from InfoTrac OneFile.

American Library Association. (2006, Jan-Feb). Chapter 6: Innovative uses of the OpenURL. *Library Technology Reports*, 42(1), pp.35-37. Retrieved June 23, 2006, from InfoTrac OneFile.

Dahl, M. (2004, February). OpenURL. *Computers in Libraries*, 24(2), p.26. Retrieved June 23, 2006, from Academic Search Premier.

Sources Consulted



- Machovec, G. & Stockton, M. (2004). OpenURL link resolvers: a practical approach with Gold Rush. *Technical Services Quarterly*, 21(4), pp. 1-16. Retrieved June 24, 2006, from Haworth Press Journals at <http://www.haworthpress.com>
- Van de Sompel, H. & Hochstenbach, P. (1999, April). Reference linking in a hybrid library environment: part 1: frameworks for linking. *D-Lib Magazine*, 5(4). Retrieved June 26, 2006, from http://www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt1.html
- Van de Sompel, H. & Hochstenbach, P. (1999, April). Reference linking in a hybrid library environment: part 2: SFX, a generic linking solution. *D-Lib Magazine*, 5(4). Retrieved June 26, 2006, from http://www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt2.html
- Van de Sompel, H. & Beit-Arie, O. (2001, March). Open linking in the scholarly information environment using the OpenURL framework. *D-Lib Magazine*, 7(3). Retrieved June 26, 2006, from <http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html>
- Van de Sompel, H. & Beit-Arie, O. (2001, July/August). Generalizing the OpenURL framework beyond references to scholarly works: the Bison-Futé model. *D-Lib Magazine*, 7(7/8). Retrieved June 26, 2006, from <http://www.dlib.org/dlib/july01/vandesompel/07vandesompel.html>