

DE STANDAARD

OpenURL en CoiNS

Theo van Veen

[leader]

Open URL is een standaard om metadata in een URL op te nemen. Applicaties – zogenaamde link resolvers – gebruiken de informatie in zo'n OpenURL om de gebruiker door te verwijzen naar de voor hem geschikte en toegankelijke bronnen. COinS biedt een methode om metadata op een gestandaardiseerde manier in een HTML-pagina te zetten, zodat er gemakkelijk OpenURLs in de HTML mee gegenereerd kunnen worden.

OpenURL

Stel: een gebruiker heeft in een database van een instelling, bijvoorbeeld in Picarta, een beschrijving gevonden van een tijdschriftartikel. Hij wil dat artikel graag inzien en zijn bibliotheek heeft dat artikel misschien wel zelf in huis. Nu zou Picarta voor elk tijdschriftartikel links kunnen opnemen naar de volledige tekst van elk artikel, bij elke bibliotheek, en vervolgens op grond van de locatie van de gebruiker de juiste link kunnen aanbieden naar de volledige tekst van het artikel waar deze gebruiker toegang toe heeft. Het is echter veel eenvoudiger om op basis van de metadata, zoals issn, volume, jaar en pagina, een standaard-link, een OpenURL, te genereren. Deze link bestaat dan uit een basis-URL, die specifiek is voor de bibliotheek van de gebruiker met daarachter de gestandaardiseerde metadata. Die basis-URL is het internetadres van de applicatie bij de desbetreffende bibliotheek die een dergelijke URL met metadata kan verwerken. Deze OpenURL ziet er dan bijvoorbeeld zo uit:

<http://www.bib.nl/resolver?genre=article&issn=12345678&volume=1&year=2007&issue=1&page=10>

Hierin is www.bib.nl/resolver de basis-URL, en het gedeelte achter het vraagteken identificeert het desbetreffende artikel door middel van metadata. De applicatie – een zogenaamde link resolver - moet de OpenURL vertalen naar de daadwerkelijke URL van het desbetreffende artikel.

Op bovenstaande manier hebben Pica en de UB Utrecht het idee van een generieke URL al eind jaren negentig in praktijk gebracht voor de koppeling van beschrijvingen in Picarta naar de beschikbare fulltext bij de UB Utrecht. In diezelfde tijd is ook het concept van OpenURL bedacht en ontwikkeld door Herbert van de Sompel van de universiteitsbibliotheek van Gent. Samen met het bedrijf Ex Libris werd het uitgewerkt tot een standaard, en Ex Libris baseerde er een commerciële implementatie op: SFX, wat staat voor *special effects*.

Het mechanisme werd niet alleen gebruikt vanuit de catalogus maar ook voor databases met tijdschriftartikelen op CDROM. In de bijbehorende software kon de basis-URL van de eigen *fulltext*-database worden geconfigureerd en via die basis-URL konden gebruikers vervolgens vanuit de beschrijving van een artikel op de CDROM naar de volledige tekst bij de eigen instelling worden geleid.

Uiteraard hoeft een en ander zich niet tot de identificatie van tijdschriftartikelen te beperken. Andere metadata, zoals 'auteur' en 'titel' kunnen ook in een OpenURL worden meegegeven, en het beschreven object mag ook een boek of iets anders zijn. Behalve het ophalen van de volledige tekst, kan de OpenURL resolver op basis van deze metadata ook allerlei andere links genereren, zoals een zoekactie op ISBN bij Amazon of op auteur in Google. In veel gevallen doen bestaande link resolvers dat ook: ze bieden een lijst aan van links die gebaseerd zijn op de metadata uit de OpenURL. Deze links moeten dus ook afhankelijk zijn van de context: voor de metadata van een boek dienen andere links aangeboden dan voor de metadata van een tijdschriftartikel. De aangeboden links kunnen ook afhankelijk zijn van de toegangsrechten van de gebruiker. De link resolver moet dan wel over een database beschikken (bijvoorbeeld de "Knowledge Base" van SFX) waarin opgezocht kan worden of de organisatie of de gebruiker toegang heeft tot een specifiek tijdschrift. Men spreekt in dit verband wel van de '*appropriate copy*'.

Eén probleem is hierbij nog niet helemaal opgelost. Hoe weet de aanbieder van de metadata het adres van de link resolver (de basis-URL) die voor een bepaalde gebruiker van toepassing is? Hiervoor zijn verschillende oplossingen. In het voorbeeld van Picarta is dat adres van de verschillende bibliotheken bij Pica bekend, en Pica kan op grond van het internetadres van de gebruiker vaststellen bij welke bibliotheek die gebruiker is aangesloten. Een andere oplossing is die waarbij de gebruiker in staat gesteld wordt dit adres zelf op te geven. De database-aanbieder slaat het dan op in een cookie op de pc van de gebruiker. Bij elke volgende sessie wordt dit adres weer uit het cookie uitgelezen en met de metadata tot een OpenURL gecombineerd. Een derde mogelijkheid is dat de database-aanbieder in een register het internetadres van de gebruiker opzoekt in een database van OCLC waarin voor een groot aantal bibliotheken de bijbehorende basis-URL te vinden is.

Open URL is door de NISO (*National Information Standards Organisation*) als ANSI-standaard Z39.88 overgenomen. Er zijn twee versies: 0.1 is de oorspronkelijke laagdrempelige versie. OpenURL 1.0 is de nieuwe versie waarbij de verzameling metadata een 'Context Object' genoemd wordt.

COinS

Om OpenURLs te kunnen genereren en zichtbaar te maken vanuit een gepresenteerde HTML-pagina, is vervolgens COinS ontwikkeld. Coins betekent *Context Objects in Spans*. Het is een standaard manier om metadata in een HTML-pagina te plaatsen. Het doel ervan is om het met behulp van eenvoudige toevoegingen aan de browser mogelijk te maken om OpenURL links in een gepresenteerde HTML pagina te genereren en zichtbaar te maken. De voor deze OpenURLs gebruikte basis-URL is daarbij afhankelijk van de gebruiker of de instelling waar de gebruiker deel van uitmaakt. Het COinS concept is begin 2005 gegroeid uit discussies op de gcs-pcs discussielijst.¹

Om het idee achter COinS uit te leggen, maken we eerst een zijstapje naar Google Scholar zoals dat ruim twee jaar geleden beschikbaar kwam. In Google Scholar zijn veel tijdschriftartikelen opgenomen. De metadata daarvan kunnen gebruikt worden om een artikel op te zoeken. Om dat proces voor de gebruiker eenvoudiger te maken, kan er aan de browser van de gebruiker software toegevoegd worden om de metadata in de HTML te herkennen en met die metadata een OpenURL te genereren. Deze OpenURL kan als link worden toegevoegd aan de Google Scholar pagina zoals die op dat moment in de browser van de gebruiker beschikbaar is. Een mechanisme hiervoor is het gebruik van "bookmarklets". Dit zijn stukjes javascript die als koppeling in de browser worden opgenomen. Klikkt men op die koppeling dan wordt dit stukje javascript uitgevoerd. Om te zien hoe dit werkt, volgt u dit instructies in het kader.

KADER

Degenen die bekend genoeg zijn met het beheer van favorieten in hun browser kunnen het idee van een bookmarklet zelf uitproberen door bij de 'koppelingen' een een nieuwe koppeling toe te voegen. Hoe dat moet, verschilt in de verschillende browsers en browserversies. In Internet Explorer 6.0: maak 'koppelingen' zichtbaar via 'beeld' > 'werkbalken' > 'koppelingen'. Die koppelingen zijn nu als knop met de term 'links' te zien, rechts van de adresbalk. Klik met de rechtermuisknop op een willekeurige link in dit lijstje, en selecteer 'eigenschappen'. Knip en plak het volgende stukje javascripts naar het veld 'URL' onder het tabblad 'webdocument':

```
javascript:txt=""; for (i=0 ; i <document.links.length ; i++) txt+=document.links[i]+'\\n'; alert(txt)
```

Geef desgewenst onder tabblad 'Algemeen' de koppeling een andere naam, en klik op OK. Als u nu op de nieuw aangemaakte koppeling klikt, wordt dit stukje javascript uitgevoerd. Het effect is dat U alle links ziet verschijnen van de op dat moment gepresenteerde HTML pagina. Het resultaat heeft in dit geval geen praktisch nut, maar illustreert alleen hoe via een koppeling of 'bookmarklet' een stukje javascript kan worden uitgevoerd. Om de links in bijvoorbeeld Google Scholar te herkennen en op basis daarvan een link naar een eigen link resolver toe te voegen is een ingewikkelder stuk javascript nodig. Dit zou hier te ver voeren maar om een en ander in werking te zien kunt U de link volgen die bij de uitleg van CoinS in dit artikel wordt gegeven.

BIJSCRIFT KADER: zelf een 'bookmarklet' maken.

In Firefox is het zelfs mogelijk zo'n stukje javascript altijd actief te laten zijn en dit automatisch te doen. Zo'n stukje javascript wordt een *browser extensie* genoemd.

Google heeft het overigens mogelijk gemaakt om het adres van de OpenURL van de eigen instelling in Google Scholar op te geven en op grond daarvan een OpenURL aan te bieden.

Het hierboven beschreven mechanisme zou men natuurlijk het liefst voor meerdere websites willen kunnen toepassen. Het is echter niet handig als je voor iedere website een aparte browserextensie moet hebben. Bovendien hoeft een aanbieder maar iets aan de presentatie te wijzigen en het mechanisme werkt niet meer. Het vergt dus ook veel onderhoud. Het is dus wenselijk de metadata op een gestandaardiseerde manier in de HTML pagina op te nemen zodat men met dezelfde browser extensie voor meerdere vergelijkbare websites kan gebruiken. Dat is waar COinS om de hoek komt kijken: metadata kunnen als Context Object (dit is een container

¹ <http://cipolo.med.yale.edu/mailman/listinfo/gcs-pcs-list> 'For folks interested in developments and developing the "Gather, Create, Share" and "Personal Collection Systems" memes, and systems implementing either or both'.

voor OpenURL metadata) in een webpagina worden vastgelegd. Aardig in dit verband is dat COinS ook ontstaan is vanuit de ontwikkeling van een browser extensie om vanuit Google Scholar een OpenURL te genereren.

Hoe werkt COinS?

Hoe COinS werkt, wordt hier uitgelegd aan de hand van een sterk vereenvoudigd voorbeeld.

In een HTML-pagina kunnen stukken tekst gemarkeerd worden met HTML-elementen. Een van die elementen is het `` element. Dit wordt gebruikt om de gemarkeerde tekst op een bepaalde manier te presenteren. Zonder opmaak-informatie ziet dat er uit als:

```
<span>Shakespeare</span>
```

Aan dit element kunnen attributen worden toegevoegd, bijvoorbeeld de kleur waarmee de gemarkeerde tekst gepresenteerd moet worden:

```
<span color="red">Shakespeare</span>
```

Met COinS wordt nu het hele Context Object in het attribuut 'title' van het `` element gezet. Om deze span als "Context Object" herkenbaar te maken wordt ook een 'class' attribuut toegevoegd met als waarde "Z3988" (de naam van de officiële standaard). Dit ziet er dan uit als:

```
<span class="Z3988" title="isbn=12345678">Hamlet</span>
```

De gebruiker ziet dit Context Object niet, maar ziet alleen de tekst 'Hamlet'. Met een eenvoudig stukje javascript is het nu mogelijk alle spans met `class="Z3988"` te herkennen en met de gegevens uit het ContextObject een OpenURL te vormen en de gemarkeerde tekst aanklikbaar te maken met die OpenURL als link. In werkelijkheid ziet het Context Object in de span er aanzienlijk complexer uit. Het bevat meestal veel meer gegevens, in een voor mensen niet bepaald makkelijk leesbare vorm:

```
<span class="Z3988" title="ctx_ver=Z39.88-2004&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.isn=1045-4438"></span>
```

Om COinS echt in werking te zien, zie: <http://curtis.med.yale.edu/dchud/resolvable/>.

Het concept van OpenURL en CoinS biedt veel perspectief. De nogal complexe Context Objects kunnen ook gevuld worden met eenvoudige metadata-formaten. Men spreekt in dit verband wel van 'micro formats'.

Als men eenmaal een browser extensie heeft die OpenURL's in een HTML pagina kan genereren, dan kan die browser extensie ook een deel van het werk van de OpenURL resolver doen en directe links naar de gewenste resources aanbieden. Het aardige hiervan is weer dat de gebruiker zelf zijn browser extensies kan kiezen en de regie kan krijgen over de services die hem voor de Context Objecten ter beschikking gesteld worden.

Verder lezen

OpenURL: http://www.niso.org/standards/standard_detail.cfm?std_id=783 en

<http://en.wikipedia.org/wiki/OpenURL>.

COinS: <http://ocoins.info>.