Koninklijke Bibliotheek

Nieuwe gegevensarchitectuur ondersteunt nieuwe diensten
Deel 2

Theo van Veen en Paul Doorenbosch

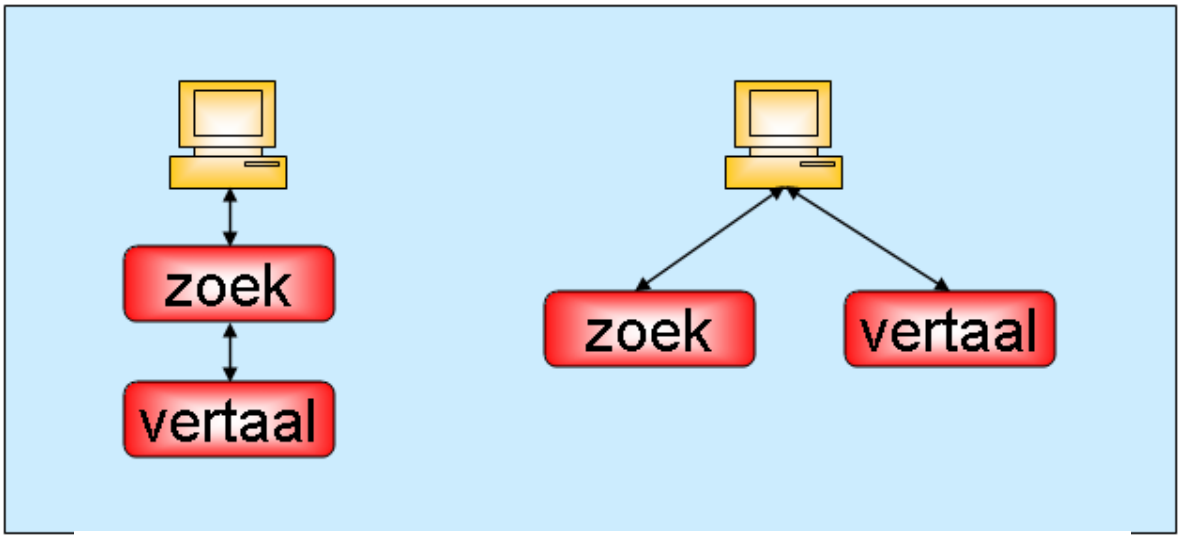
De KB heeft haar gegevensinfrastructuur vernieuwd. De service-georiënteerde architectuur die hieraan ten grondslag ligt, maakt het onder andere mogelijk om eenvoudig en snel diensten te realiseren met de data van de KB en die van anderen. In het vorige nummer van IP werd de nieuwe gegevensarchitectuur beschreven. Hier wordt uitgelegd hoe hiervan gebruik gemaakt kan worden om web 2.0-functionaliteit te bieden.

Stel U vindt een abstract van een tijdschriftartikel in een taal die U niet spreekt maar U vermoedt dat het artikel interessant is. Zou het niet handig zijn als U die samenvatting direct vertaald te zien kreeg? Of misschien wilt U bij het zoeken in een bibliotheekcatalogus altijd wel direct zien of een boek verfilmd is. Kortom U wilt misschien dat sommige dingen automatisch gepresenteerd worden. Echter, het wordt niet aangeboden. U moet de juiste website opzoeken en zelf met knippen en plakken de gegevens invoeren om de door U gewenste informatie te krijgen.In dit artikel wordt uitgelegd hoe dit automatisch gedaan kan worden.
Data en diensten moeten met een minimum aan inspanning geïntegreerd kunnen worden met die van andere partijen. Met integratie wordt hier bedoeld dat de output van een webservice gebruikt kan worden als input voor andere webservices en zonder dat die services onderling van elkaar afhankelijk zijn. Op dit moment zijn gebruikers afhankelijk van de functionaliteit die een aanbieder van een dienst in de gebruikersinterface aanbiedt. Sommige gebruikers willen echter bruikbare services van andere partijen op een eenvoudige manier, d.w.z. automatisch of met één muisklik, kunnen integreren met resultaten uit bijvoorbeeld zoekacties in een willekeurige database.

In dit artikel wordt aan de hand van sterk vereenvoudigde voorbeelden uitgelegd hoe een service georiënteerde infrastructuur en standaardisatie van de gegevensinfrastructuur de basis kunnen vormen voor deze integratie. Mogelijk dat de gemiddelde gebruiker hier niet direct gebruik van maakt, maar hetgeen hier beschreven wordt kan ook aantrekkelijk zijn voor de aanbieder van een website omdat een site aantrekkelijk gemaakt kan worden met de diensten van anderen.

Het architectuurmodel

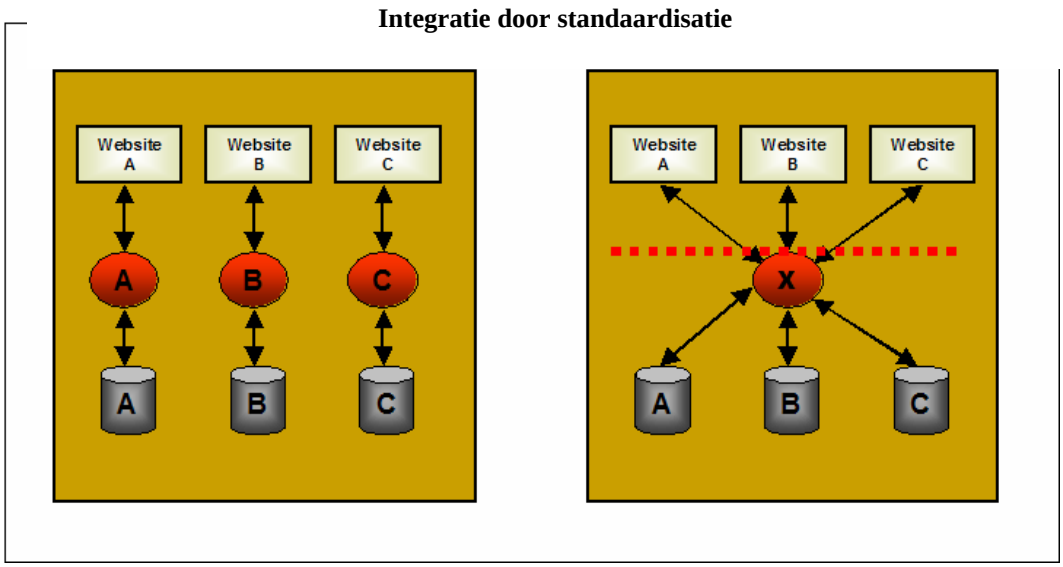
Twee archtectuur modellen



Figuur 1 Links voorbeeld van integratie die in de dienst is vastgelegd. Rechts is de integratie onder controle van de gebruikersapplicatie.

Het concept van een servicegeoriënteerde architectuur wordt uitgelegd aan de hand van een eenvoudig voorbeeld. De huidige praktijk is meestal dat bij het benaderen van een dienst de integratiemogelijkheden met andere diensten bepaald worden door de aanbieder van een dienst of de leverancier van de software. Dit is geschetst in het linker deel van figuur 1 met als voorbeeld een zoekdienst en vertaaldienst. De gebruiker zoekt gegevens in een database en de aanbieder van de dienst biedt de mogelijkheid om de resultaten te vertalen. Het kan ook anders. Rechts in figuur 1 zien we de situatie waarbij de vertaalservice direct vanuit het werkstation van de gebruiker wordt aangeroepen. De integratie van de twee diensten, zoeken en vertalen, vindt hier plaats op het werkstation van de gebruiker.
Deze sterk vereenvoudigde weergave laat zien dat de gebruiker vrij kan zijn in de keuze van de vertaalservice als hij de verwijzing naar de vertaalservice zelf zou kunnen veranderen. Sterker nog: de gebruiker kan de mogelijkheid geboden worden willekeurige en van elkaar onafhankelijke diensten naar eigen believen koppelen. Deze diensten hoeven er geen weet van te hebben dat hun output gebruikt wordt als input voor een andere dienst.
Om deze integratie te kunnen realiseren moet de webapplicatie (bijvoorbeeld een bibliotheekcatalogus) waarmee de gebruiker gezocht heeft wel de mogelijkheid bieden om de verwijzingen naar andere diensten te beïnvloeden. De applicaties waarin deze integratiemogelijkheden geboden worden zijn de *clients*. De diensten die hiermee benaderd worden noemen we de *services*.

Hoe kunnen we services nu integreren? Allereerst is het nodig dat de client 'weet' hoe de verschillende services benaderd moeten worden, dat wil zeggen hoe een verzoek geformuleerd wordt of hoe de URL eruit moet zien, en hoe de output geïnterpreteerd moet worden. Het is dan ook wenselijk dat deze twee aspecten voor gelijksoortige services zoveel mogelijk hetzelfde zijn en dat daarvoor dezelfde regels gelden. We spreken in dit geval van een protocol. Voor het kunnen interpreteren van de output is het wenselijk dat deze volgens bekende schema's (dataformaten) wordt opgebouwd.



Figuur 2 Links is de huidige praktijk weergegeven: elke website benadert een specifieke database via een eigen protocol, A, B of C. Rechts de situatie waarin alle webdiensten alle databases kunnen benaderen door hetzelfde protocol (protocol X) te gebruiken.

In figuur 2 is weergegeven hoe de integratie bevorderd wordt, indien verschillende diensten op dezelfde manier benaderd kunnen worden. Links zien we de situatie dat elke website alleen maar gebruik kan maken van een eigen lokale service of database (bijvoorbeeld een bibliotheekcatalogus) omdat eigen lokale protocollen en dataformaten gebruikt worden die niet door andere websites ondersteund of gekend worden. Rechts zien we dat meerdere websites verschillende services of databases kunnen benaderen omdat door standaardisatie hetzelfde protocol en dataformaat gebruikt wordt. Het zal duidelijk zijn dat standaardisatie de integratie bevordert omdat het voor de client niet uitmaakt waar de data vandaan komen.
Voor Search en Retrieval is de rechter situatie nu realiseerbaar omdat we daarvoor het standaard SRU protocol (Search and Retrieval via URL's) en standaard formaten zoals Dublin Core[1] kunnen gebruiken. Het gebruik van een standaard dataformaat maakt het nu ook eenvoudiger om de verkregen data als input voor andere diensten te gebruiken. Het maakt immers in dat geval niet uit van wie de data afkomstig zijn. Echter, niet voor alle diensten zijn standaarden vastgesteld. We moeten de wereld echter nemen zoals die is en niet wachten tot alles gestandaardiseerd is. Om een client toch in staat te stellen automatisch niet-gestandaardiseerde services te gebruiken moeten we de client van voldoende informatie voorzien over die services. Dit doen we d.m.v. servicebeschrijvingen in machine-leesbare vorm. Hiermee wordt een client in staat gesteld die services 'automatisch' te benaderen en de output op de juiste manier te interpreteren. Automatisch betekent hier dat de client een service kan aanbieden afhankelijk van de context. Verder moet de client hieruit voldoende informatie kunnen halen om het resultaat van de service te presenteren.

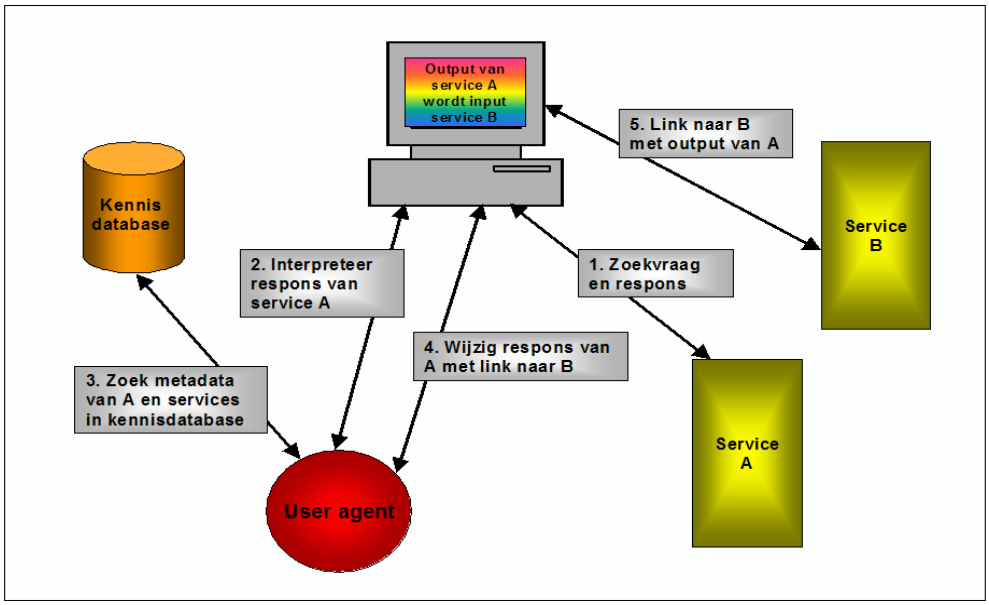
Service-integratie

Om het idee van service-integratie te verduidelijken gaan we uit van een situatie met de volgende componenten:

- Een webpagina met bijvoorbeeld het resultaat van een zoekactie. Dit is het startpunt voor de gebruiker.

- Een stukje programmatuur (user agent) dat in staat is de webpagina te interpreteren en eventueel te modificeren. Dit kan bijvoorbeeld een onderdeel zijn van een portal of een uitbreiding van de browser. De internetbrowser Firefox biedt faciliteiten voor dit soort uitbreidingen. Bij een portal is de user agent zonder speciale actie van de gebruiker beschikbaar. In het tweede geval moet de gebruiker die uitbreiding zelf installeren.
- Services. Een service kan in principe elke webapplicatie zijn die via een URL aangesproken wordt. Google kan dus ook als service gezien worden.
- Een kennisdatabase met beschrijvingen van relevante services. Deze servicebeschrijvingen bevatten o.a. informatie over hoe een service aangesproken wordt (de syntax van de URL) en welke criteria aanleiding zouden moeten geven deze service aan de gebruiker aan te bieden. Als er bijvoorbeeld een ISBN getoond wordt kan dat aanleiding zijn om een link naar Amazon aan te bieden. Ook moet vastgelegd zijn wat het type output is en

Servive integratie



Figuur 3. Schematische weergave van de stappen om met behulp van servicebeschrijvingen twee services te integreren.

  hoe deze output van de service verwerkt moet worden. Als de output een afbeelding van de boekomslag is kan deze in de beschrijving getoond worden. Is de output HTML dan zal een link geboden moeten worden.

In figuur 3 is het scenario weergegeven waarbij bovengenoemde componenten gebruikt worden. De opeenvolgende stappen zijn genummerd aangeven. De gebruiker komt op een webpagina van service A met bijvoorbeeld een zoekresultaat (1). De user agent interpreteert de webpagina en vindt daarin bepaalde gegevens, bijvoorbeeld een auteursnaam (2). De user agent controleert in de kennisdatabase welke services, op grond van het voorkomen van het veld auteursnaam, aan de gebruiker aangeboden zouden kunnen worden (3). Veronderstel dat dit service B is. In dat geval verandert de user agent de auteursnaam in een link naar de service B met de auteursnaam ingevuld op de juiste positie in de URL van deze link (4). De gebruiker heeft nu de mogelijkheid naar service B door te klikken met de

auteursnaam automatisch in de URL ingevuld (5).

De user agent heeft nu twee van elkaar onafhankelijke services geïntegreerd op het niveau van de presentatie, dus zonder dat er in de services zelf ingegrepen hoefde te worden en zonder dat die services iets van elkaar of van die user agent weten. Indien de gebruiker invloed heeft op de kennisdatabase, kan deze integratie onder controle van de gebruiker gebracht worden.

Er is hier echter nog wel een probleem: hoe kan de user agent de webpagina interpreteren? Omdat in webpagina's doorgaans de metadata niet als zodanig herkenbaar zijn is dit lastig. We zullen daar verderop op ingaan. Indien een zoekresultaat in XML beschikbaar gemaakt kan worden is het herkennen van metadata wel mogelijk.

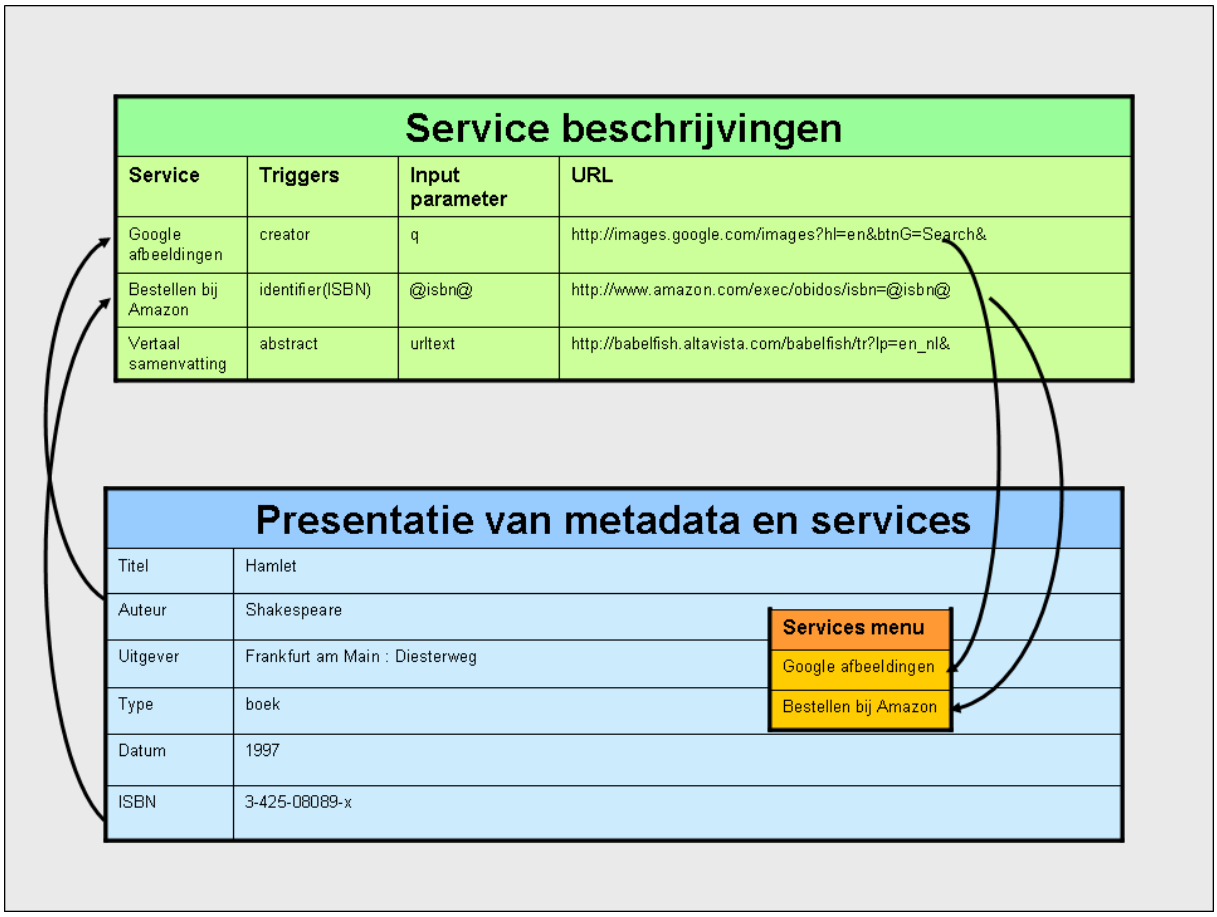
Demonstratie- en testportal

We kunnen het concept van service integratie illustreren aan de hand van een werkende demo[2]. Het gaat hier om een experimentele portal bij de KB die via het SRU protocol gelijktijdig in meerdere databases kan zoeken. Met deze demo portal kan gezocht worden in bibliografische en tekstbestanden.
De portal is gebaseerd is op Ajax-technologie. Ajax staat voor Asynchronous, Javascript en XML en wil zeggen dat vanuit een webpagina door middel van Javascript XML-pagina's opgevraagd kunnen worden bij verschillende servers. Asynchronous houdt hier in dat de resultaten verwerkt kunnen worden zodra deze ontvangen zijn en zonder dat het scherm "bevriest" tijdens het wachten op de nog niet ontvangen resultaten.
Het Ajax-concept leent zich uitstekend voor het integreren van services. In deze demo wordt hiervan gebruik gemaakt door gelijktijdig zoekopdrachten te versturen naar verschillende SRU-services. In feite is dit de situatie die rechts in figuur 2 is geschetst omdat voor al die services hetzelfde protocol en hetzelfde dataformaat gebruikt wordt. Door het gebruik van Ajax-technologie kan het benaderen van services op de achtergrond gebeuren en pas om gebruikersinteractie vragen indien de respons van een service daartoe aanleiding geeft. Door het gebruik van XML kan de client (portal) de ontvangen data zelf in het gepresenteerde resultaat inpassen. In deze demo wordt het Ajax concept nu overigens nog alleen gebruikt voor het zoeken via SRU, maar dit zal in de toekomst uitgebreid kunnen worden met andere services die output in XML leveren.

De zoekresultaten zijn metadata volgens Dublin Core in XML formaat. Een XML-file met servicebeschrijvingen dient als kennisdatabase. Voor elke service wordt hierin aangegeven welke Dublin Core metadatavelden in de respons aanleiding moeten geven voor het aanbieden van deze service. Verder is het adres van de service en de bijbehorende URL-syntax erin vastgelegd. Het concept wordt in sterk vereenvoudigde vorm geïllustreerd in figuur 4. Hierin zijn de service-beschrijvingen te zien van "Google afbeeldingen", "Bestellen bij Amazon" en "Vertaal samenvatting". De kolom "triggers" geeft aan bij welke metadata velden deze services geboden moeten worden. In dit geval wordt op grond van het voorkomen van 'creator' de link "Google afbeeldingen" gegenereerd, voor 'ISBN' de link "Bestellen bij Amazon" en voor "abstract de link "Vertaal samnevatting". Kijken we nu naar de presentatie van de metadata dan zien we een services-menu met zowel "Google afbeeldingen" en

“Bestellen bij Amazon”. De link “Vertaal samenvatting” wordt niet aangeboden omdat in de metadata het veld “abstract” niet voorkomt.
De input parameter in de derde kolom geeft aan in welke URL-parameter het metadata veld ingevuld en vervolgens toegevoegd moet worden aan de URL voor die service. Bij Google is dat de parameter “q”; bij Amazon is er niet zo’n parameter en is in de servicebeschrijving aangegeven dat @isbn@ vervangen moet worden door de waarde van het ISBN. Het laatste veld in de laatste kolom geeft de URL van de service waar het desbetreffende veld aan toegevoegd of ingevuld moet worden.

Service beschrijvingen en gepresenteerde services



Figuur 4 Schematische weergave van de relatie tussen servicebeschrijving en het presenteren van metadata. De metadatavelden auteur (creator) en isbn (identifier) worden in de servicebeschrijving als trigger genoemd en dat leidt hier tot het tonen van een menu met links naar de bijbehorende services.

In bovenstaand voorbeeld blijkt het nut van standaardisatie van de achterliggende gegevensinfrastructuur. De metadata-velden die hier van belang zijn, worden weliswaar getoond met eigen labels maar de onderliggende data zijn Dublin Core velden. Het bovenstaande werkt daarom ook voor andere SRU services die de data in Dublin Core aanbieden. Echter ook andere applicaties dan een SRU portal kunnen van die service beschrijvingen gebruik maken. Dit vraagt dus ook om standaardisatie van die servicebeschrijvingen.

In de praktijk zullen de servicebeschrijvingen ingewikkelder zijn dan de vereenvoudigde weergave hierboven. Ten eerste zijn er meer manieren om services te benaderen dan alleen via de parameters in een URL. Ook kan de output heel

verschillend van aard zijn (HTML, XML, images etc). Het zou te ver voeren om op deze aspecten in te gaan. Bovendien is het bepalen van het beste schema voor het goed beschrijven van alle relevante aspecten van een service op een zodanige manier dat verschillende typen applicaties deze goed kunnen gebruiken nog onderwerp van onderzoek.

De demonstratie-portal biedt de gebruiker de mogelijkheid een eigen file met servicebeschrijvingen te laden zodat deze vrij is in de keuze van services. Uiteraard zal een gemiddelde gebruiker niet zelf servicebeschrijvingen aanmaken. Echter, “gevorderde” gebruikers kan de mogelijkheid geboden worden servicebeschrijvingen te maken en aan te bieden zodat deze gedeeld kunnen worden met andere gebruikers.

Het mag uit het bovenstaande duidelijk zijn dat het concept service hier ruim genomen moet worden en niet beperkt is tot bijvoorbeeld webservices gebaseerd op SOAP[3], het protocol om computertoepassingen op gestandaardiseerde wijze via het web met elkaar te laten communiceren. De voorbeelden betroffen hier overigens alleen services die als link worden aangeboden en een HTML pagina opleveren. Er zijn echter talrijke toekomstige uitbreidingen denkbaar, waarbij de input voor een service niet beperkt is tot metadata. Ook een image of video zou als input kunnen dienen. Een andere denkbare uitbreidingen is dat de output geïntegreerd wordt met het oorspronkelijke zoekresultaat. Om een idee te geven: in plaats van een link aan te bieden naar een vertaalservice kunnen metadata direct vertaald gepresenteerd worden. Of stel dat de metadata een URL bevatten naar een video dan zou een service denkbaar zijn die deze video van ondertiteling voorziet. Het zal duidelijk zijn dat voor dit soort toepassingen het schema voor de service beschrijvingen verder onderzoekt vereist om deze generiek genoeg te maken om al deze mogelijkheden te dekken.

Relatie met OpenURL, COinS en microformats

Het idee om vanuit een bibliografische beschrijving andere diensten aan te roepen kan gezien worden als een vervolg op het OpenURL[4] -concept. Bij OpenURL worden metadata als parameters in een URL geplaatst. Deze URL verwijst naar een zogenaamde link resolver (in feite de user agent) die op basis van deze URL een HTML pagina aanmaakt met links naar diverse diensten, afhankelijk van de beschikbare metadata. De aangeboden diensten zijn gebaseerd op een kennisdatabase met daarin gegevens over die diensten. Deze kennisdatabase is meestal afhankelijk van de leverancier of de aanbieder van de linkresolver (bijvoorbeeld de SFX linkresolver van Ex Libris).

In het in dit artikel geschetste concept wordt deze link resolver overgeslagen en worden direct dynamische en contextafhankelijke links in de presentatie van het zoekresultaat aangeboden. Bovendien is de kennisdatabase vervangen door servicebeschrijvingen die in principe door de gebruiker uit te breiden en te veranderen zijn. Standaardisatie van deze service beschrijvingen wordt nagestreefd om deze uitwisselbaar te maken en dus niet leveranciersafhankelijk te laten zijn.

Een uitbreiding, die zorgt dat service integratie niet beperkt blijft tot gebruik binnen een portal, is om metadata in gewone webpagina's op te nemen. Een concept hiervoor is COinS (Context Object in Span).[5] Hierbij wordt in de HTML in een zogenaamde "span" alle relevante context informatie (bijv. auteur, ISBN etc.) vastgelegd. Een user agent (in dit geval een browser extensie) kan deze informatie in de HTML detecteren en op basis daarvan de pagina modificeren en nieuwe links aanbieden.

Dit kan geïllustreerd worden in een sterk vereenvoudigde vorm van COinS. Stel dat een auteur in een HTML pagina wordt opgenomen als:

```
Dit werk is geschreven door <span title="dc:creator">Shakespeare</span>
```

Indien hier niets mee gebeurt, dan ziet de gebruiker gewoon de tekst:

```
Dit werk is geschreven door Shakespeare
```

Met behulp van een browser extensie kunnen dit soort "spans" opgezocht worden in de webpagina en op basis van de servicebeschrijvingen veranderd worden in een aanklikbare link naar bijvoorbeeld "Google afbeeldingen" met "Shakespeare" als parameter:

Dit werk is geschreven door Shakespeare

Waarbij Shakespeare een link is naar:
http://www.google.nl/imghp?hl=nl&tab=wi&q=shakespeare
Klikken op deze link levert dan de afbeeldingen van Shakespeare.

Bij COinS ziet de span er overigens veel ingewikkelder uit dan hierboven omdat COinS eigenlijk bedoeld is om alle metadata te bevatten die een object beschrijven plus nog een aantal andere gegevens.

Zouden alle metadata in een HTML pagina beschikbaar gesteld worden met markeringen zoals in het voorbeeld in het kader dan spreken we van semantic tagging. Een user agent zou ook hier de HTML pagina kunnen verrijken met links naar door de gebruiker geselecteerde services. Bij zo'n vorm van semantic tagging spreken we van "Microformats"[6]. Een user agent zou hiervoor dezelfde servicebeschrijvingen kunnen gebruiken als in het voorbeeld van de demo portal.

Services bij de KB

Bij de Koninklijke Bibliotheek is de nieuwe gegevensinfrastructuur nu bijna voltooid. Alle websites en alle diensten maken gebruik van Dublin Core met waar nodig uitbreidingen. Er is een begin gemaakt met de servicegeoriënteerde architectuur, d.w.z. dat nieuwe diensten nu zo veel mogelijk in de vorm van losse services gerealiseerd worden. Waar in het verleden vaak sprake was van monolithische pakketten, wordt bij nieuwe projecten nu steeds meer uitgegaan van hergebruik van bestaande services en geïntegreerde toegang.

Voorbeelden van services in de nieuwe infrastructuur zijn de SRU-services voor zoeken, OAI-services voor harvesting en een “resolutie” service die zowel standaard identifiers als interne KB identifiers vertaalt naar een internet locatie. Andere voorbeelden zijn een service om op afbeeldingen in te zoomen en een service om tekst in de images van gedigitaliseerde tekst te “highlighten”. Ook de aanvraagfunctie voor uitleenbaar materiaal zal zo worden ingericht dat deze vanuit verschillende diensten kan worden opgeroepen.
Waar dat toegestaan is, zal langzamerhand gebruik gemaakt worden van bestaande services van derden, zoals het tonen van de afbeeldingen van voorpagina’s van boeken en het tonen van gegevens uit bijvoorbeeld Worldcat om de gebruiker te laten zien wat de dichts bijzijnde locatie van een boek is. Een nieuwe zoekingang voor de KB catalogus, waarin deze nieuwe functionaliteit zichtbaar is, is bijna klaar. Het standaardiseren van servicebeschrijvingen is nog in de onderzoeksfase en zal een onderdeel worden van een werkpakket van een nog te starten Europees project (TELplus). De ideeën rond de personalisatie zoals te zien in de demo portal zijn nog niet uitgekristalliseerd.

Toekomst

Het in dit artikel beschreven concept laat zien wat de mogelijkheden van web 2.0 zien. De basis hiervoor wordt gelegd door standaardisatie van metadata, de manier waarop gegevens worden uitgewisseld en de manier waarop services beschreven worden. Hoewel wat in dit artikel beschreven is deels nog in een onderzoeksfase is, lijkt het voor gebruikers en organisaties perspectieven te bieden om gebruik te maken van functionaliteit die al door anderen gerealiseerd en beschikbaar gesteld is. Door integratie van die functionaliteit met eigen diensten kan de dienstverlening aan de gebruikers aanzienlijk te uitgebreid worden.

Met enige fantasie kan men zich voorstellen dat het hier beschreven concept zeer veelbelovend kan zijn voor toekomstige ontwikkelingen. Zo kan het aanbieden van een service gebeuren op grond van veel complexere criteria dan hier beschreven, zoals het voorkomen van een combinatie van metadata eventueel of juist het ontbreken van metadata of het terugkrijgen van een leeg zoekresultaat. En uiteraard kan het concept uitgebreid worden tot het automatisch aanbieden van een aaneenschakeling van services, bijvoorbeeld een samenvatting wordt eerst vertaald en vervolgens omgezet in gesproken tekst.
Door het standaardiseren van de servicebeschrijvingen zijn deze niet gekoppeld aan een specifieke toepassing en wordt het mogelijk ze onderling uit te wisselen. Zodoende worden ze ook voor applicaties van derden bruikbaar. Indien instellingen hun servicebeschrijvingen op een standaard manier publiceren (bijvoorbeeld in een centrale registry) komen deze services ook voor anderen beschikbaar. Deze servicebeschrijvingen kunnen dan doorzocht worden en gebruikers kunnen deze toevoegen aan een persoonlijke kennisdatabase. Deze persoonlijke database kan dan weer beschikbaar gesteld worden aan applicaties (user agents) die dit concept geïmplementeerd hebben.

Kortom, door beschikbare services op de juiste wijze te beschrijven, kunnen intelligente applicaties hiermee hun weg op het Internet zoeken. Deze applicaties kunnen dan datgene automatisch en eventueel op de achtergrond doen wat de

gebruiker anders zelf zou moeten doen, of niet zou doen vanwege de hoeveelheid werk. De toegevoegde waarde voor de gebruiker kan dus enorm groot zijn.

Theo van Veen is projectadviseur bij de Hoofdafdeling Research and Development van de Koninklijke Bibliotheek.

Paul Doorenbosch is hoofd van de afdeling product- en dienstonwikkeling van de Koninklijke Bibliotheek.

---

[1] SRU werd besproken in De Standaard, IP 2007(4), p. 32-35; Dublin Core in IP 2006(?), p. xx-xx OPZOEKEN

[2] De demoportal is te vinden op: http://research.kb.nl/sruportal

[3] SOAP/Simple Object Access Protocol, http://en.wikipedia.org/wiki/SOAP

[4] The OpenURL Framework for Context-Sensitive Services, http://www.niso.org/committees/committee_ax.html

[5] OpenURL COinS (A Convention to Embed Bibliographic Metadata in HTML), http://ocoins.info/
OpenSearch en COinS werden besproken in De Standaard in IP 2007(3), p. 32-34

[6] Microformats, http://microformats.org/