

Scheiden van XML-data en presentatie

Omzetten van XML naar HTML, SVG en PDF

XML wordt alom beschouwd als dé standaard voor het uitwisselen van gegevens en speelt een cruciale rol in business-to-business (B2B) en application-to-application (A2A) integratie. XML in combinatie met XSL Transformations (XSLT) kan gebruikt worden om op een effectieve manier data en presentatie van elkaar te scheiden. In dit artikel wordt beschreven hoe XML en XSLT gebruikt kunnen worden om flexibele en krachtige web-applicaties te ontwikkelen op het Oracle9i-platform voor het ontsluiten van informatie.

XML in combinatie met XSLT resulteert in verhoogde flexibiliteit en verminderde onderhoudskosten van applicaties. Door gebruik te maken van XSLT kunnen XML-documenten – die alleen data bevatten – getransformeerd worden naar presentatieformaten zoals HTML voor presentatie in een normale browser, WML voor presentatie op een WAP-toestel en SVG voor het presenteren van informatie in grafieken. XML-documenten kunnen ook omgezet worden naar PDF voor professionele presentatie en afdrukken.

XML

XML is een meta-taal voor het markeren (formatteren) van data. Een XML-document bevat alleen maar data. Dit in tegenstelling tot een HTML-document dat ook markeringen (*markup*) bevat voor presentatie in een browser. De XML-standaard schrijft niet voor hoe bijvoorbeeld een order, vluchtschema of nieuwsbericht in XML gemarkeerd moet worden. Deze XML-talen (of *vocabulaires*) moeten ontwikkeld worden door industrieën, consortia of bedrijven. Het voorbeeld hieronder is een XML document dat de data bevat van een order met drie orderregels.

```
<?xml version="1.0"?>
<order reference="A-156-3" customer_id="3827984">
  <items>
    <item article_id="125817" amount="30"/>
    <item article_id="368763" amount="10"/>
    <item article_id="324674" amount="20"/>
  </items>
</order>
```

De data in een XML-document zijn bevat in *elementen* en *attributen*. Een element begint met een *start tag* (bijvoorbeeld `<items>`) en eindigt met een *end tag* (bijvoorbeeld `</items>`). Alles wat tussen deze tags staat is de *content* van dat element. Het element 'items' in bovenstaand voorbeeld bevat drie *subelementen* (één voor iedere orderregel). Attributen bevinden zich binnen de start tag van een element en bestaan uit een naam en een waarde (bijvoorbeeld `article_id="125817"`). Het vocabulaire van het bovenstaande XML document is gedefinieerd in een DTD (*Document Type Definition*) of *XML Schema* en beschrijft de structuur, elementen en attributen die gebruikt worden om de ordergegevens te markeren.

XML-documenten kunnen *data centric* zijn en business objecten bevatten zoals orders en vluchtschema's, of *document centric* zijn en bijvoorbeeld nieuwsberichten, e-mails of helpteksten bevatten. Data centric documenten worden in de regel gegenereerd door systemen en zijn bedoeld voor *machine consumption* (verwerking door andere systemen). Daarentegen worden document centric documenten meestal geschreven door een persoon en zijn deze bedoeld voor *human consumption* (presentatie aan andere personen).

Eén van de belangrijkste toepassingen van XML is het uitwisselen van gegevens in B2B- en A2A-integratieoplossingen. XML is daarnaast de fundering van web services technologieën zoals SOAP, WSDL en UDDI. Dit artikel richt zich op een andere belangrijke toepassing van XML: het presenteren van informatie.

XSLT

De XSL (Extensible Stylesheet Language) bestaat uit twee delen: XSLT en XSL Formatting Objects (XSL FO). De tweede wordt later in dit artikel behandeld. XSLT is een XML-taal waarmee beschreven wordt hoe een XML-document in vocabulaire A getransformeerd moet worden naar een XML-document in vocabulaire B. Zoals eerder gezegd schrijft de XML-standaard niet voor hoe orders, vluchtschema's of nieuwsberichten in XML gemarkeerd moeten worden. Dit heeft geresul-

teerd in een groot aantal vocabulaires ontwikkeld door industrieën, consortia en bedrijven. Sommige van deze vocabulaires beschrijven (bijna) dezelfde informatie maar gebruiken een afwijkende structuur of afwijkende elementen en attributen. Het XML document hieronder bevat dezelfde data als het eerder gegeven voorbeeld maar heeft een andere structuur.

```
<?xml version="1.0"?>
<order>
  <cust>3827984</cust>
  <custref>A-156-3</custref>
  <orderline>
    <prodid>125817</prodid>
    <amount>30</amount>
  </orderline>
  <orderline>
    <prodid>368763</prodid>
    <amount>10</amount>
  </orderline>
  <orderline>
    <prodid>324674</prodid>
    <amount>20</amount>
  </orderline>
</order>
```

XSLT kan worden toegepast om het XML-document in vocabulaire A te transformeren naar het XML-document in vocabulaire B. Door XSLT te gebruiken kan een systeem, dat gebouwd is om orders in vocabulaire A te verwerken, orders in vocabulaire B ontvangen, transformeren naar vocabulaire A en vervolgens verwerken. Deze toepassing van XSLT speelt een cruciale rol in veel B2B- en A2A-integratieoplossingen. XSLT is ook de technologie die het mogelijk maakt om data en presentatie te scheiden. 'Vocabulaire B' kan namelijk een XML-taal zijn die beschrijft hoe data gepresenteerd moet worden, zoals (X)HTML, WML, SVG en XSL FO. Door een XML-document, dat alleen maar data bevat, met behulp van een XSLT stylesheet te transformeren naar een XML-presentatieformaat worden data en presentatie gescheiden.

Voordelen van scheiden data en presentatie

Veel webapplicaties hebben programma's (bijvoorbeeld Java Servlets) die dynamisch HTML pagina's aanmaken. Meestal is de logica (code) om de data op te halen en de logica om de data te formatteren in HTML met elkaar verweven. Door XML en XSLT toe te passen kan de logica op een nette manier van elkaar gescheiden worden. Namelijk door eerst de data op te halen en te markeren in XML en vervolgens de XML-data met behulp van XSLT te transformeren naar HTML. Deze opzet verbetert de onderhoudbaarheid omdat de presentatielogica aangepast kan worden zonder de logica aan te hoeven passen die de data ophaalt. Een ander voordeel is dat beide stukken logica parallel aan elkaar ontwikkeld kunnen worden. Na

afstemming van het XML-formaat kan een ontwikkelaar werken aan de logica om de data op te halen, terwijl een andere ontwikkelaar de presentatielogica in XSLT schrijft.

Personalisatie

Deze toepassing hoeft zich niet te beperken tot een enkele HTML-presentatie. Meerdere HTML-presentaties van dezelfde XML-data kunnen gebruikt worden voor gepersonaliseerde presentatie. Aan gebruikers met verschillende rollen of met verschillende informatiebehoefte, kunnen verschillende presentaties van dezelfde XML-data aangeboden worden. Zo'n gepersonaliseerde presentatie kan ook gebaseerd zijn op voorkeuren van gebruikers (kleuren, fonts en dergelijke). De verschillende presentaties worden gerealiseerd door verschillende XSLT stylesheets toe te passen.

Meerdere formaten

Ook hoeft de toepassing zich niet te beperken tot een enkel formaat. Bij *multi-channel publishing* wordt dezelfde XML-data naar verschillende XML-presentatieformaten getransformeerd afhankelijk van het toestel (device) waarmee de applicatie benaderd wordt: HTML voor een reguliere browser, WML (WAP Markup Language) voor WAP-toestellen en een hele reeks van andere formaten voor andere mobiele toestellen zoals PDA's.

XSQL Server Pages

XSQL Server Pages is een zeer toegankelijke en productieve technologie die onderdeel uitmaakt van de Oracle XML Developer's Kit (XDK). Een XSQL Server Page is een XML-document dat één of meer SQL queries en instructies om de resultaten van de queries in XML te formatteren bevat. Hieronder is een voorbeeld van een XSQL Server Page waarmee een lijst van eilanden wordt opgehaald. Van ieder eiland worden de volgende gegevens opgehaald: land, totale (land)oppervlakte, lengte van de kustlijn, bevolkingsdichtheid en afstand tot het dichtstbijzijnde continent.

```
<?xml version="1.0"?>
<xsql:query xmlns:xsql="urn:oracle-xsql"
  connection="demo"
  rowset-element="islands"
  row-element="island"
  id-attribute=""
  tag-case="lower">
  SELECT i.name
  ,      c.name as country
  ,      i.total_area
  ,      i.shoreline
  ,      i.pop_density
  ,      i.dist_continent
FROM    islands i
  ,      countries c
WHERE   i.country_id = c.id
ORDER  BY i.name
</xsql:query>
```

Een XSQL Server Page heeft de extensie 'xsql' en kan benaderd worden via een browser. Om dit te kunnen doen moet de Oracle Application Server zodanig geconfigureerd worden dat bij het benaderen van een XSQL Server Page het XSQL Servlet gestart wordt. Het Servlet leest de XSQL Server Page, voert deze uit en geeft de resultaten terug aan de browser. Het 'connection'-attribuut refereert aan een database connectie die is gedefinieerd in de 'XSQLConfig.xml'-configuratiefile van het XSQL Servlet. De attributen 'rowset-element', 'row-element', 'id-attribute' en 'tag-case' zijn instructies om de XML-data op een bepaalde manier te formatteren. Als deze instructies niet zijn opgenomen dan wordt de XML-data op een standaard manier (het zogenaamde *canonical format*) geformatteerd. In figuur 1 is het XML-document te zien dat in de browser getoond wordt bij het benaderen van de XSQL Server Page.



Figuur 1. Resultaat XSQL Server Page

Door aan de XSQL Server Page de onderstaande processing instruction toe te voegen kunnen we het XSQL Servlet instrueren om een transformatie uit te voeren.

```
<?xml-stylesheet type="text/xsl" href="islands_html.xsl"?>
```

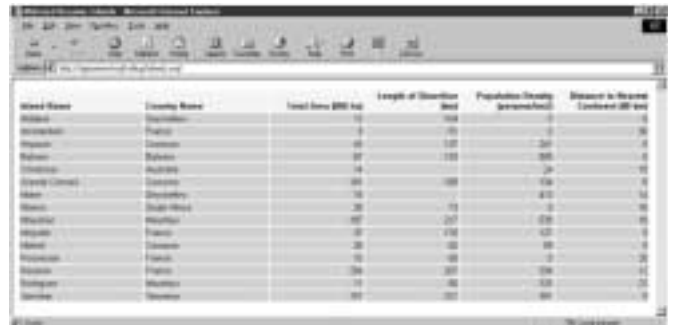
Wanneer we vervolgens opnieuw de XSQL Server Page benaderen wordt een HTML-pagina getoond waarin de informatie gepresenteerd wordt. Zie figuur 2.

Presentatie in grafieken

Bij veel systemen zijn er vereisten om informatie in grafieken te tonen. HTML zelf biedt hier weinig mogelijkheden voor. Scalable Vector Graphics (SVG) is een XML-taal die in combinatie met HTML gebruikt kan worden voor onder andere het tonen van grafieken. SVG is een krachtige taal waarmee twee-

Succesvolle samenwerking is afhankelijk van zowel 'zachte' als 'harde' factoren

dimensionale grafische afbeeldingen beschreven kunnen worden. Deze afbeeldingen zijn opgebouwd uit drie typen objecten: vormen (lijnen, rechthoeken, etc.), tekst en images. SVG-afbeeldingen kunnen dynamisch en interactief zijn. Met SVG kunnen hele mooie resultaten bereikt worden als je SVG goed kent. Maar ook met beperkte kennis van SVG kunnen goede resultaten behaald worden. Om een SVG-afbeelding te kunnen zien in een browser moet een SVG viewer als plug-in geïnstalleerd zijn. Van Adobe kan een gratis viewer gedownload worden.



Figuur 2. HTML presentatie.

In figuur 3 is het resultaat te zien van de transformatie van de XML-data uit figuur 1 naar SVG. Omdat de SVG-afbeelding opgebouwd is uit vectoren kan de gebruiker inzoomen zonder dat de afbeelding vervormt.



Figuur 3. SVG presentatie

Island Name	Country Name	Year Area (km²)	Length of Shoreline (km)	Population	Distance to Nearest Continent
Aldabra	Seychelles	43	127	0	100
Ascension	Britain	4	11	0	100
Bahamas	Bahamas	43	127	297	10
Bahamas	Bahamas	34	180	288	10
Chagos Islands	Maldives	14	180	0	10
Crozet Islands	France	201	180	114	10
Madagascar	Madagascar	59	142	15	10
Madagascar	Madagascar	157	217	12	10
Madagascar	Madagascar	57	142	12	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10
Madagascar	Madagascar	35	65	35	10

Figuur 4. PDF presentatie

```

<?xml version="1.0"?>
<svg xml:space="preserve" width="1000" height="1000">
  <desc>Mid-sized Oceanic Islands</desc>
  <g style="stroke:#000000;stroke-width:1;font-family:Arial;font-size:16">
    <rect x="200" y="55" height="25" width="13" style="fill:#000080"/>
    <text x="20" y="73">Aldabra</text>
    <text x="223" y="73">13</text>
    <rect x="200" y="90" height="25" width="9" style="fill:#000080"/>
    <text x="20" y="108">Amsterdam</text>
    <text x="219" y="108">9</text>
    ...
  </g>
</svg>

```

In dit voorbeeld is de SVG-afbeelding zelf in de browser getoond. Een SVG-afbeelding kan echter ook opgenomen worden in een HTML-pagina. Dit kan simpelweg door een image op te nemen waarbij in het 'src'-attribuut van het 'img'-element verwezen wordt naar de URL van de SVG-afbeelding.

Gebruik van PDF-formaat

Een vervelende eigenschap van een HTML-pagina is dat deze niet goed afgedrukt kan worden. In veel gevallen valt er tekst weg aan de rechterkant van de pagina. Als professionele afdrucken gewenst zijn voldoet HTML meestal niet. PDF is een veelgebruikt professioneel afdrukformaat dat ook prima in een browser getoond kan worden. Hieronder wordt uitgelegd hoe XML-data kan worden getransformeerd naar PDF.

XSL Formatting Objects

Zoals al eerder beschreven bestaat XSL uit twee delen: XSLT en XSL Formatting Objects (XSL FO). De eerste hebben we al eerder behandeld in dit artikel. XSL FO is een XML-taal waarmee in detail beschreven kan worden hoe een document moet

worden afgedrukt. De taal is zeer uitgebreid en biedt veel mogelijkheden. Omdat het zo een uitgebreide taal is en omdat het World Wide Web Consortium (W3C) haar specificaties op een nogal academische wijze opstelt is XSL FO lastig te doorgronden. Het is niet mogelijk om de resultaten van een transformatie naar XSL FO direct in een browser te tonen. Eerst zal het XSL FO-document naar een formaat moeten worden omgezet dat kan worden getoond in een browser of afgedrukt. Eén van de mogelijkheden is het omzetten naar PDF.

Formatting Objects Processor

Apache biedt een utility voor het omzetten van XSL FO naar PDF: de Formatting Objects Processor (FOP). Deze utility kan XSL FO omzetten naar diverse formaten maar wordt vooral gebruikt voor PDF. De utility is geschreven in Java en kan geïnstalleerd worden door een aantal Java resources te downloaden en deze te plaatsen in het Java class path dat gebruikt wordt bij het uitvoeren van het Java Servlet waar vanuit de FOP-utility aangeroepen wordt (bijvoorbeeld het XSQL Servlet). Omdat de FOP utility de Xalan XML Parser van Apache gebruikt moet ook deze gedownload worden en opgenomen worden in hetzelfde Java class path. Vanuit een XSQL Server Page is het heel eenvoudig om de FOP utility te gebruiken. Hiervoor hoeft slechts het 'serializer' attribuut toegevoegd te worden aan de onderstaande processing instruction.

```

<?xml-stylesheet type="text/xsl" href="islands_fo.xsl"
serializer="FOP"?>

```

Door het opnemen van dit attribuut zal de FOP-utility het XSL FO-document dat het resultaat is van de transformatie automatisch omzetten naar PDF. De browser zal vervolgens de PDF plug-in gebruiken om het PDF document te tonen. Zie figuur 4.

De toepassing van transformaties naar XSL FO in combinatie met FOP biedt een krachtig alternatief voor Oracle Reports voor professionele printuitvoer. In een web-applicatie kan bij een HTML presentatie een optie geboden worden om de informatie in PDF te tonen om af te drukken. Bij het gebruik van XML en XSLT kunnen deze presentaties op dezelfde XML-data gebaseerd zijn.

Erwin Groenendal

is technisch directeur van Cumquat Information Technology. Hij heeft tien jaar ervaring met Oracle technologie. E-mail: erwin.groenendal@cumquat.nl.