

XML en XSLT met rdbms in de praktijk

Nuon heeft succes met systeem voor nota-presentatie

Jaron Borensztajn en Bart van Oort

Inzet van XML- en XSLT-technieken in databases maakt het mogelijk via een internetbrowser documenten te tonen en te printen. Voor bedrijven die zeer veel documenten versturen levert deze oplossing toegevoegde waarde voor klanten. PTT Post gebruikt XML op deze wijze voor haar klant Nuon. Vragen over facturen van dit energiebedrijf zijn nu eenvoudig te beantwoorden. De applicatie is gebaseerd op een functioneel framework van componenten. Een praktijkcase.

Beantwoorden van vragen over facturen, of het opnieuw versturen van de energierekening aan de klant - het was in het verleden een complexe bezigheid. Met de applicatie Postview van Privver, een onderdeel van PTT Post, kan Nuon nu de facturen presenteren voor het callcenter en opnieuw printen en versturen. De in de browser getoonde factuur is een exacte kopie van de fysiek geprinte nota; zie figuur 1. Dit biedt grote voordelen voor de communicatie tussen de medewerker in het callcenter en de klant. Aanpassingen aan het factureringssysteem van Nuon zijn hiervoor niet nodig.



FIGUUR 1: VOORBEELD VAN EEN FACTUUR GETOOND IN EEN BROWSER.

Voordat het mogelijk is documenten te tonen of te printen, is een heel proces doorlopen. Dezelfde printstroom die het energiebedrijf aanlevert bij PTT Post wordt tevens ingelezen in Privver Postview. Het gehele proces -van het inlezen van de printstroom in Privver Postview tot het kunnen tonen en uitprinten van individuele facturen vanuit een browser- verloopt met behulp van een geïntegreerde toepassing van een rdbms, XML en XSLT.

XML: een definitie

Volgens een definitie van de Universiteit Utrecht is XML (eXtensible Markup Language) een computertaal die beschrijft hoe met behulp van een bepaald soort codering verschillende structurele en inhoudelijke aspecten van teksten zijn te markeren. Kennis over de tekst gaat daardoor deel uitmaken van de tekst zelf. Hierdoor ontstaan zelfbeschrijvende teksten, die tot op zekere hoogte automatisch zijn te verwerken. Een belangrijk uitgangspunt is dat de markeringen alleen betrekking hebben op de inhoud van de tekst, en niet op de bedoelde weergave van de tekst in een bepaald medium (op papier, op het Web of nog anders). Door de scheiding van informatie-inhoud en weergave wordt het mogelijk dezelfde informatie op verschillende manieren te presenteren - afhankelijk van de voorkeur van het publiek, of van de eisen van het medium.

XSLT

XSLT (eXtensible Stylesheet Language - Transformations) wordt vanuit de Domstad omschreven als een taal voor het transformeren van XML-documenten in nieuwe XML-documenten dan wel in html. Zogeheten XSLT-templates geven aan hoe de coderingen worden omgezet in leesbare informatie. De vertaalslag naar html is uiteraard van belang in omgevingen waarin documenten bij voorkeur worden geraadpleegd met behulp van internetbrowsers.

COMPONENTEN-ARCHITECTUUR

De toepassing van Privver Postview is gebaseerd op een functioneel framework voor *electronic bill presentment and payment* (EBPP). Dit raamwerk bestaat uit technische bouwblokken die gezamenlijk de databaselaag, de logica en de presentatielaag verzorgen. Bovenop de technische componenten ligt een laag met functionele

componenten op een hoog niveau, zoals een XML-converter en een workflow-component. De Postview-applicatie is opgebouwd uit een aantal modules uit het framework.

In een systeem met miljoenen documenten is performance altijd een belangrijk issue

Een groot voordeel van deze modulaire architectuur is dat de toepassing in korte tijd kon worden ontwikkeld en dat door de componentenstructuur eenvoudig wijzigingen en uitbreidingen zijn aan te brengen. Figuur 2 geeft een overzicht van de applicatie-architectuur.

PRINTSTROOM ALS INPUT

Printstromen vormen de input voor Privver Postview. Een printstroom bestaat uit de elektronische verzameling facturen of andere documenten die bedrijven aanbieden aan een printbedrijf, in dit geval Print en Mail, eveneens een onderdeel van PTT Post. Gingen printstromen tot voor kort rechtstreeks de printer in, nu worden ze afgetapt en als input gebruikt voor elektronische distributie via Privver Postview. Tijdens de verwerking worden de documenten eerst geconverteerd naar XML. Deze conversie is complex. Men transformeert printgegevens naar logische gegevens. Daarnaast voegt de verwerker extra XML-tags toe, om de achterliggende applicaties met de bijbehorende functionaliteiten te ondersteunen. De XML-omzetting voor een nieuwe printstroom vindt plaats via een initiële configuratie. Daarbij wordt de structuur van de printstroom ontarfeld in vaste (lay-out-)elementen en variabele elementen, zoals naam, adres, notabedrag en omschrijving.

Elke printstroom -bijvoorbeeld nota's van Nuon- gebruikt een specifieke stylesheet. Zo is de lay-out van de documenten gescheiden van de variabele velden in het document. Daardoor blijft de benodigde opslagcapaciteit relatief beperkt.

Bovendien is het mogelijk om voor verschillende toepassingen andere 'views' op het document te definiëren door een andere stylesheet te kiezen. De documenten die na splitsing en conversie overblijven, worden in XML-formaat opgeslagen in een omvangrijke relationele database. In dit geval is SQL Server gebruikt. De onderliggende componenten ondersteunen tevens Oracle. In de database kan snel een query worden aangemaakt op bijvoorbeeld klantnummer, klantnaam, factuurnummer of andere documentkenmerken.

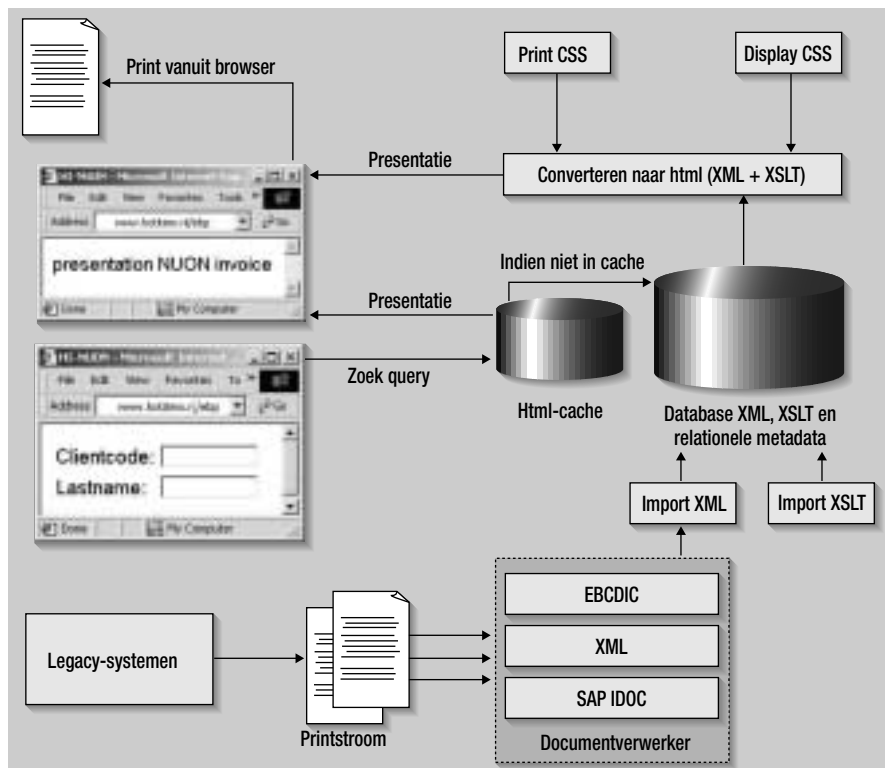
In de html-conversie wordt het opgeslagen XML-document met behulp van XSLT en de bij de printstroom horende stylesheet omgezet naar html en vervolgens op het scherm gezet. De view op het scherm is bij gebruik van de standaard XSLT-stylesheet identiek aan de originele geprinte factuur of het document. Een andere mogelijkheid is het gebruik van een aangepaste stylesheet voor specifieke toepassingen, zoals het printen van het document of het aanbieden van een doorsnede van het document aan de gebruiker.

Voor Nuon is een speciale interface gebouwd die de nota's, afkomstig uit het SAP-systeem van het bedrijf, converteert naar XML. Dankzij de converter van

IDOC naar XML die nu beschikbaar is, hebben klanten als Nuon geen speciale koppeling met het SAP-systeem nodig.

TECHNISCHE UITDAGINGEN

Het ontwikkelteam werd geconfronteerd met een aantal technische uitdagingen. Op enkele daarvan gaan we nader in. Allereerst geven we antwoord op de vraag waarom een relationele database is aangewend voor het opslaan van XML. Die keuze heeft alles te maken met de hoeveelheid informatie. De totale database bevat honderdduizenden documenten en zal op termijn groeien tot vele miljoenen documenten. Bij de opzet is speciaal op performance gelet. Daarom is de database opgesplitst in een statisch en een dynamisch gedeelte. Zoeken kan in een kleine dynamische database, waarin alleen de sleutelvelden staan met een verwijzing naar een documentnummer. De grote database met documenten heeft slechts één sleutel, namelijk het documentnummer. Doordat de grote documentendatabase nauwelijks wijzigt, is met database-partitionering aan de hand van datum een zeer goede performance te realiseren.



FIGUUR 2: APPLICATIE-ARCHITECTUUR PRIVVER POSTVIEW.

Extra document-views

Door het gebruik van een aangepaste XSLT-stylesheet is het mogelijk een 'view' op een document te maken. Deze zijn door middel van autorisatie toe te wijzen aan gebruikers. Door een view te creëren en te autoriseren, kun je bepaalde gedeelten van een document -bijvoorbeeld bedragen- afschermen voor een bepaalde groep gebruikers. Een andere toepassing is het toevoegen van een geheel andere kleurstelling of lay-out. Tevens laat de document-view zich gebruiken voor nabewerking in de printstroom, om elementen aan het document toe te voegen of juist te verwijderen. Je kunt op die manier een tekstgedeelte aan een nota toevoegen zonder verandering van de aanleverende applicatie, zoals de helptekst voor de callcenter-agent.

Bovendien is het gemakkelijker om bijvoorbeeld oude documenten ouder dan een jaar in bulk te verwijderen of te verplaatsen naar een ander archief.

Tijdens het traject bleek dat de conversie van XML-documenten naar html met behulp van XSLT-stylesheets veel processorvermogen vereist - een potentiële flessenhals. Dit werd ontdekt tijdens het opvragen van documenten. De oplossing lag in caching van de html-pagina's. Dit is flexibel in te stellen, op basis van beslisregels. Beslisregels zijn bijvoorbeeld het aantal dagen dat een document in de cache wordt bewaard of de grootte van het XML-document. Nuon heeft ervoor gekozen recentelijk opgevraagde documenten en alle nieuw binnengekomen documenten in de database van tevoren te converteren naar html. Het geheel gaat vervolgens in de html-cache. Je kunt immers verwachten dat klanten vragen zullen stellen over de rekeningen of brieven die zij het laatst hebben ontvangen.

Een correcte reproductie van de pagina-overgangen in een XML-document bleek geen sinecure. Hiervoor werd in de XML-converter speciaal een regelteller bijgehouden, die na een aantal regels expliciet een pagina-overgang-tag toevoegt in het XML-document. Op die manier ziet de opgevraagde pagina er precies hetzelfde uit als de geprinte versie.

BUSINESS-VOORDELEN

Veel bedrijven die grote hoeveelheden documenten verwerken, hebben baat bij een XML/XSLT-gebaseerde oplossing. De hier beschreven applicatie is bedoeld voor bedrijven die veel post versturen. Een specifieke toepassing voor bedrijven met een callcenter is Privver Postview. Hiermee kunnen vragen over facturen worden beantwoord door het online tonen van verstuurd nota's en het uitprinten van een specifieke rekening. De output die nu naar de printstraat wordt gestuurd, dient direct als input voor de applicatie. Dit betekent

dat de printgegevens zich als input laten gebruiken en dat geen specifieke interface nodig is. Internationale bedrijven met facturen of documenten in uiteenlopende talen en meertalige callcenters kunnen garen spinnen bij het gebruik van XML in de database. Met de op de XML gebaseerde beschreven methode, is het mogelijk het presenteren en uitprinten van documenten in meerdere talen te ondersteunen. Hiervoor hoeft in het aanleverende systeem geen meertaligheid te zijn ingebouwd; alleen het callcenter moet de gewenste taal hebben ingesteld.

Gebruik van XSLT zorgt voor flexibiliteit in de lay-out. Bedrijven kunnen verschillende typen facturen tonen met het aanleveren

XSLT biedt flexibiliteit in de lay-out

van dezelfde printstromen. Aanpassing aan de documentverwerker is hier niet nodig. De enige conversie bestaat uit de aanmaak van een nieuwe XSLT-stylesheet. Bij elk type factuur behoort een eigen XSLT-stylesheet, bijvoorbeeld: de euro. Aanlevering van de data blijft onveranderd en het tonen van een andere lay-out is mogelijk door aanpassing van de sheet.

BETERE DIENSTVERLENING

Een combinatie van XML, XSLT met een rdbms is uitstekend geschikt voor het elektronisch verwerken van documenten. Door een dergelijk systeem te bouwen op basis van een EBPP-functioneel framework krijgt de klant binnen korte tijd een op zijn wensen afgestemde applicatie.

Tabel 1 geeft een overzicht van de voordelen van de gebruikte technieken. Toepassing van relatief jonge ontwikkelingen zorgt in dit geval voor het online digitaal presenteren van fysieke documenten. Het callcenter bereikt hiermee een hoger niveau van dienstverlening aan de klant; zowel kwantitatief -kortere gesprekken- als kwalitatief: beide gesprekspartners hebben exact dezelfde lay-out voor zich, wat leidt tot minder irritaties. Deze voordelen rechtvaardigen de investering. ●

Jaron Borensztajn (borensztajn@hotitem.nl) is applicatie-architect bij HotITem Informatica. In die functie is hij verantwoordelijk voor de consistentie binnen applicaties en het hergebruik van componenten binnen de diverse door dit bedrijf ontwikkelde applicaties, waaronder Privver Postview. Bart van Oort (oort@hotitem.nl) is functioneel ontwerper bij Hot ITem Informatica. Hij heeft voor het EBPP-framework diverse klantwensen vertaald naar webgebaseerde functionele modules.

XML/XSLT	Relationele database	Functioneel framework voor EBPP op basis van componenten
Vormgeving	Structuur van textuele data	Schaalbaarheid
Verschillende views	Snel zoeken	Mogelijkheid tot uitbreiden functionaliteit
Meertaligheid	Metadata-opslag	Architectuur
Flexibiliteit	Snelle performance door caching	Gedegen beheer

TABEL 1: OVERZICHT VAN DE VOORDELEN VAN DE GEBRUIKTE TECHNIKEN IN PRIVVER POSTVIEW.