

De rol van XML in Multi-tier Client/Server Architectuur

XML alleen biedt voor Business-2-Business een onvoldoende basis

In relatief korte tijd is XML uitgegroeid tot de nieuwe haarlemmerolie van de IT-industrie. Elke zichzelf respecterende leverancier van ontwikkelomgevingen, databases, webservers of applicatieservers doet iets met XML in een poging mee te liften op de grote aandacht van klanten voor XML, al is het vaak niet meer dan het veelvuldig noemen van de drieletterige afkorting in brochures en presentaties. Dit artikel probeert enerzijds een overzicht te verschaffen in de toepassingsmogelijkheden van XML aan de hand van de multi-tier client/server architectuur. Anderzijds wordt gepoogd het realistisch beeld te schetsen van de mogelijkheden en onmogelijkheden van XML in diverse toepassingsmogelijkheden aan de hand van een eenvoudig hulpmiddel: de XML-Stack.

Multi-tier Client/Server Architectuur

Door een aantal duidelijk te onderscheiden trends in het bedrijfsleven heeft de multi-tier client/server architectuur zich de laatste jaren ontwikkeld als dé bouwstijl [1]. Enerzijds hebben trends als globalisatie, internationale samenwerkingsverbanden en fusies en overnamen ervoor gezorgd dat bedrijven zich voor de taak gesteld zagen hun automatiseringssystemen samen te voegen. Daarnaast heeft ook de veranderende maatschappij bijgedragen aan de noodzaak tot aanpassing van traditionele automatiseringssystemen. De 24-uurs economie heeft geen tijd meer voor 'batch-windows', producten hebben een snellere 'time-to-market' nodig dan voorheen, en nieuwere en meer diverse distributiekanaal vereisen een andere vorm van ondersteuning dan een traditionele terminal-client kan bieden. Neem als voorbeeld de verkoop van verzekeringsproducten in het filiaal van een grootgrutter of via het Internet.

Klassieke monolithische applicaties (stovepipes) lenen zich doorgaans slecht voor specifieke functionele ontsluiting door de aard van hun architectuur, aangezien dataontsluiting, applicatiefunctieeliteit en presentatielogica niet gescheiden zijn. Het is juist deze scheiding die de multi-tier client/server architectuur bepaalt.

In de technologische visie van de multi-tier client/server architectuur wordt gedefinieerd dat applicaties verdeeld worden in tenminste een drietal fysieke lagen: client, applicatieserver en enterpriseserver.

De client-laag bevat de presentatielogica van de applicatie. Deze presentatie kan gerealiseerd worden middels een traditionele programmeertaal en worden getoond in een traditionele Grafische User Interface (GUI), middels internettechnologie in bijvoorbeeld een webbrowser, een Netwerk Computer, of middels WAP-technologie in een mobiele telefoon of een Personal Digital Assistant. Aan de andere kant van het spectrum bevindt zich de enterpriseserver-laag, die enerzijds de gegevensverzameling van de applicatie beheert en anderzijds de bestaande (legacy) systemen. Ook systemen die als gevolg van samenwerkingsverbanden, fusie- en overnamepartijen binnen de scope van de IT organisatie zijn gekomen, bevinden zich logisch gezien in deze laag. De applicatieserver-laag vormt de brug tussen de client-laag en de enterpriseserver-laag door het herbergen van de applicatielogica. Deze applicatielogica werkt zowel ontsluitend naar de enterpriseserver-laag als faciliterend aan de client-laag.

Door gebruik te maken van het bovenstaande architectuurmodel wordt invulling gegeven aan de eerdergenoemde trends in het bedrijfsleven. Zo wordt globalisatie afgedekt door de mogelijkheid om op wereldwijde schaal zowel klanten (via diverse distributiekanaal) als samenwerkingsverbanden (supply chain integration) en fusies & overnames te ontsluiten. Voor deze ontsluiting kan gebruik gemaakt worden van o.a. Internettechnologie. Tevens kan hergebruik gepleegd worden van bestaande systemen. Tenslotte is er een duidelijke ontkoppeling aangebracht tussen batch en on-line applicaties waarbij de batch applicaties zich in de enterpriseserver-laag zullen bevinden. Hierdoor kan aan de voorkant (de client- en

applicatieserver-laag) 7x24 uur ondersteuning geboden worden in de frontoffice terwijl de backoffice middels batch-windows bepaalde zaken later afhandelt.

Wat is XML?

De World Wide Web Consortium (W3c) heeft de eXtensible Markup Language (XML) ontwikkeld met als doel een standaard neer te zetten voor uitwisselbare data. Bij de ontwikkeling van XML is enerzijds ontkoppeling en anderzijds uitbreidbaarheid het uitgangspunt geweest.

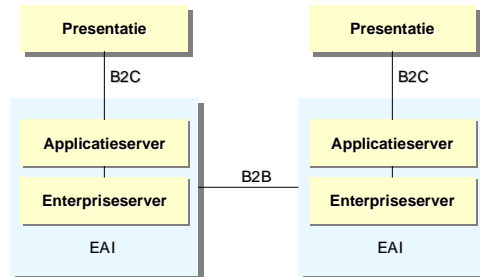
XML is, net zoals bijvoorbeeld HTML (HyperText Markup Language), een taal waarmee op gestructureerde wijze informatie vastgelegd kan worden middels zogenaamde *tags*. De tags geven de semantiek (betekenis) weer van de woorden die tussen de tags staan.

```
<Verzekerde>
  <Voornaam>Jan</Voornaam>
  <Achternaam>Janssen</Achternaam>
  <Straat>Jansingel</Straat>
  <Huisnummer>1</Huisnummer>
  <Postcode>1111 XX</Postcode>
  <Woonplaats>Amsterdam</Woonplaats>
</Verzekerde>
```

Figuur 1: XML Voorbeeld

Zo is bovenstaand voorbeeld (zie Figuur 1) de informatie die tussen de tags <Verzekerde> en </Verzekerde> staat van een verzekerde persoon, waarbij de persoon Jan Janssen uit Amsterdam betreft wonende aan de Jansingel 1 met postcode 1111 XX.

Als gekeken wordt naar de mogelijke rol die XML kan spelen in de invulling van communicatie in een multi-tier client/server architectuur, kan een drietal aspecten onderscheiden worden: business-2-consumer (B2C), business-2-business (B2B) en Enterprise Application Integration (EAI). De onderlinge samenhang tussen deze gebieden staat weergegeven in Figuur 2.



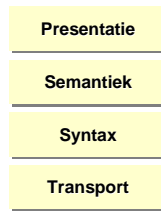
Figuur 2: Drie gebieden voor XML communicatie

XML-Stack

Om een op XML gebaseerde oplossing te definiëren, dient een aantal voorwaarden standaard ingevuld te worden. Dit zijn transport, syntax, semantiek en presentatie (zie Figuur 3). Samen vormen deze voorwaarden de XML-stack.

Transport betreft de methode waarmee XML van zender naar ontvanger wordt verplaatst. Dit kunnen standaard transportprotocollen als HTTP (het HyperText Transfer Protocol voor het verzenden en opvragen van internetpagina's) of FTP (het File Transfer Protocol voor het verzenden en opvragen van bestanden) zijn, maar evengoed een verzendmechanisme als e-mail of asynchrone middleware als IBM's MQSeries of Microsoft's MSMQ.

Syntax behelst de regelgeving rondom het juiste gebruik van tags. Enerzijds houdt dit in welke tags gebruikt mogen worden, anderzijds de wijze waarop ze gebruikt mogen worden. Hierbij valt bijvoorbeeld te denken aan regels over hoe tags genest mogen worden en dat elke "openings-tag" (bijvoorbeeld de tag <Verzekerde>) keurig afgesloten dient te worden met een "sluitings-tag" (bijvoorbeeld de tag </Verzekerde>).



Figuur 3: XML-Stack

In de laag semantiek bevindt zich informatie over de betekenis van tags. Aan welke entiteit in de werkelijkheid of een automatiseringssysteem wordt gerefereerd met de tag <adres>, of erger: met de tag <verzekerde>. Om deze vraag te beantwoorden dient een XML bericht altijd gekoppeld te zijn aan een verklarende lijst. In praktijk zijn er twee typen verklarende lijsten: de oudst bekende is de Data Type Defenitions (DTD), van recentere aard zijn de schema's, zoals Microsoft Biztalk.

Presentatie, tenslotte, bevat informatie die de uiteindelijke weergave van de gegevens in het XML bestand op een scherm of ander outputmedium bepaalt. Dit is de enige laag van de XML-stack die niet altijd ingevuld hoeft te worden, maar alleen dan als er voor een eindgebruiker iets moet worden getoond. Een voorbeeld van presentatie zijn de eXtensible Style Sheets (XSL) die worden gebruikt om aan te geven hoe informatie uit een XML bestand in een webbrowser getoond dient te worden. In de style sheet staat informatie over layout, maar ook over welk deel van de data getoond dient te worden. Zo kan van hetzelfde XML bestand een veelvoud van uiteindelijke representaties worden opgebouwd, die zowel inhoudelijk als qua inhoud van elkaar verschillen.

De XML-stack wordt in dit artikel gebruikt als hulpmiddel om aan te geven hoe de verschillende deelgebieden kunnen worden ingevuld bij ieder van de drie toepassingen van XML in een multi-tier client/server omgeving: B2B, B2C en EAI.

Business-2-Consumer

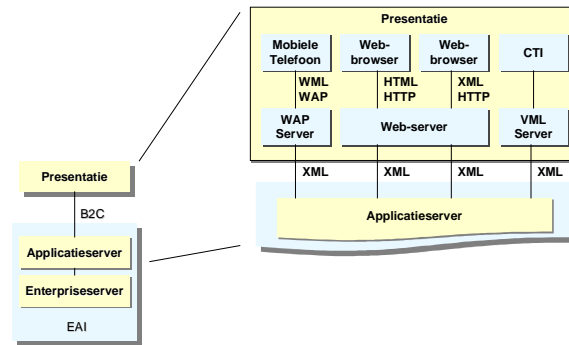
De toegevoegde waarde van XML in Business-2-Consumer communicatie is gelegen in het feit dat bedrijven steeds meer kanalen willen gebruiken om hun klanten te bedienen. Daar waar deze kanalen door automatisering worden ondersteund, is het wenselijk zoveel mogelijk dezelfde business logica te gebruiken, en per kanaal zo min mogelijk inspanning te verrichten om deze logica te ontsluiten. XML maakt deze architectuur mogelijk door als algemeen dataformaat tussen de generieke business logica en de specifieke kanaalapplicaties en apparaten te stromen.

Voor de invulling van de business-2-consumer kant van de multi-tier client/server architectuur is een drietal technologieën van belang voor de invulling van de XML-stack. Het aspect transport wordt ingevuld middels open technologieën zoals bijvoorbeeld HTTP of WAP, het aspect syntax middels XML en het aspect presentatie middels XSL.

WAP betreft het zogenaamde Wireless Access Protocol. Dit protocol is verantwoordelijk voor de communicatie met mobiele apparaten, zoals een mobiele telefoon of Palm PDA met een WAP-Browser.

XSL is een scripting taal waarmee de presentatie voor XML wordt gespecificeerd. Op deze wijze worden inhoud en presentatie gescheiden. Dit in tegenstelling tot bijvoorbeeld HTML. Door XSL aan XML toe te voegen levert een toonbaar bestandsformaat op als HTML, WML of VoXML (zie Figuurv 4).

Bovenstaande technologische invulling van de XML-stack stelt bijvoorbeeld een bankverzekeerbaar in staat om een aantal componenten te ontwikkelen waarmee klanten orders kunnen uitvoeren op de AEX en inzicht kunnen krijgen in de orderportefeuille. Als de bankverzekeraar deze componenten generiek realiseert middels een XML interface kan deze functionaliteit middels diverse technologieën aangeboden kunnen worden aan de klanten.



Figuur 4: Business-2-Consumer middels XML¹

Zo heeft de bankverzekeraar de mogelijkheid om klanten inzicht in hun orderportefeuille te geven via een mobiele telefoon. Dit doordat de door het orderportefeuille-component aangeboden informatie in XML middels XSL omgevormd kan worden naar WML en middels de WAP-Server over het WAP-protocol getransporteerd kan worden naar de mobiele telefoon.

Daarnaast kan men bijvoorbeeld dezelfde functionaliteit aan klanten aanbieden middels een webbrowser. Om inzicht te verlenen in een orderportefeuille heeft de bankverzekeraar een tweetal keuzes. Allereerst is er de mogelijkheid om de omzetting van XML naar HTML aan de kant van de verzekeraar plaats te laten vinden. Daarnaast is er de mogelijkheid om de omzetting aan de kant van de klant te laten plaatsvinden middels XSL. Hiervoor moet de klant wel de beschikking hebben over een webbrowser die zowel XML als XSL ondersteund. Het voordeel van omzetting aan de kant van de klant is dat de XML data op diverse manieren getoond kan worden middels verschillende XSL-en. Denk bijvoorbeeld aan het in een tabel tonen van de complete orderportefeuille of aan het tonen van een grafiek met de dagwaarde bij sluiting van de AEX van de orderportefeuille over de afgelopen 4 weken.

Tenslotte kan bijvoorbeeld ook dezelfde functionaliteit middels een traditionele telefoon aangeboden worden, doordat de orderportefeuille omgevormd wordt naar VoxML Voice Markup Language). Middels VML kan vervolgens met een computer omgezet worden naar spreektaal via een traditionele telefoon.

Het gebruik van XML aan de business-2-consumer kant faciliteert een drietal trends in het bedrijfsleven. Enerzijds kan middels het gebruik van de openheid van XML in combinatie met internettechnologie iedereen overal en altijd (het "Martini effect": anyplace, anytime, anywhere, ...) bediend worden (globalisatie). Daarnaast kan het bedrijfsleven middels diverse distributiekanaalen zijn producten aan de man brengen, zoals directe verkoop middels bijvoorbeeld een webbrowser of mobiele telefoon via het Internet of indirecte verkoop middels tussenhandel middels een traditionele client. Verder kan een kortere time-to-market behaald worden doordat component met een XML-interface middels diverse technologieën aangeboden kan worden of op diverse manieren gecombineerd kan worden tot nieuwe producten.

Enterprise Application Integration

Bij de invulling van de Enterprise Application Integration kant van de multi-tier client/server architectuur is een koppeling tussen systemen in de enterpriseserver-laag en in de applicatieserver-laag aan de orde. De enterpriseserver-laag kan uit een uiteenlopende verzameling bestaan. Niet alleen de gegevensverzameling van de applicatie uit de applicatieserver-laag, maar ook de bestaande (legacy) systemen en de systemen die als gevolg van samenwerkingsverbanden, fusie- en overnamepartijen binnen de scope van de IT organisatie zijn gekomen, kunnen zich in deze laag bevinden. Er kan derhalve een diversiteit aan platformen,

¹ De figuur geeft een indicatie van de mogelijkheden van XML in combinatie met andere technologieën. Er zijn legio andere mogelijkheden te bedenken om genoemde functionaliteit aan te bieden.

besturingssystemen, netwerkprotocollen en programmeertalen aanwezig zijn, die allen gekoppeld moeten worden.

Het op technisch niveau systemen koppelen van applicaties onafhankelijk van platformen, besturingssystemen, netwerkprotocollen en programmeertalen, is het gebied van middleware. Maar naast deze technische aspecten is er het probleem dat applicaties elkaar dienen te begrijpen op syntactisch en semantisch niveau. Zo maakt de ene applicatie bijvoorbeeld onderscheid tussen een straatnaam en een huisnummer, terwijl een andere applicatie dit onderscheid niet maakt. Hier biedt XML de uitkomst.

De oplossing van bovenstaande problematiek bij Enterprise Application Integration kan wederom het beste aan de hand van de XML-stack ingevuld worden.

Op de transportlaag biedt de inzet van standaard asynchrone middleware, zoals bijvoorbeeld MQSeries, BEA MessageQ of een combinatie van Microsoft MSMQ en Level 8 GenevaMQ, de oplossing om onafhankelijk platformen, besturingssystemen, netwerkprotocollen en programmeertalen een integratie tot stand te brengen. Bijkomend voordeel van het gebruik van asynchrone middleware is de ont koppeling van de frontoffice (presentatie- en applicatieserver-laag) van de backoffice (enterpriseserver-laag). Op deze wijze is het mogelijk om in de frontoffice 24-uur ondersteuning te leveren, terwijl de backoffice deze ondersteuning (nog) niet kan leveren vanwege 'batch-windows'.

Op de syntax- en semantieklaag is XML uitermate geschikt juist doordat hiermee uitwisselbare data gerealiseerd wordt die tussen de verschillende applicaties uitgewisseld kan worden. Hiervoor dient op uiteraard een uniforme afspraak gemaakt te worden over de gebruikte tags, dat wil zeggen wordt het tag <Verzekerde> of <Klant> gebruikt. Op dit vlak zijn een aantal leveranciers, onder andere Microsoft met BizTalk en HP met E-Speak, en brancheorganisaties, onder andere FinXML voor de uitwisseling van financiële gegevens, actief om standaard tags te definiëren. Daar de ontwikkeling van deze standaarden nog niet is uitgekristalliseerd, lijkt het voor de hand te liggen om een eigen bedrijfsstandaard te ontwikkelen eventueel op basis van branchestandaarden. Zo zou een verzekeringsbedrijf zijn eigen XML-standaard kunnen baseren op bijvoorbeeld de LBW-berichtenopmaak van ADN (het Assurantie DataNetwerk voor standaard berichtuitwisseling tussen tussenpersonen en verzekeringsbedrijven).

Bovenstaande technologische invulling van de XML-stack aan de Enterprise Application Integration kant van de multi-tier client/server architectuur, stelt bijvoorbeeld een telecombedrijf in staat om 24-uur per dag producten te verkopen via het Internet (de frontoffice) terwijl de feitelijke afhandeling van de aanvraag tijdens kantooruren afgehandeld wordt. Dit doordat het telecombedrijf zijn frontoffice, geïmplementeerd op basis van Internettechnologie, geïntegreerd heeft met zijn twee productgeoriënteerde systemen in de backoffice. Het ene systeem, geïmplementeerd in een Unisys omgeving, handelt productaanvragen af voor het mobiele netwerk, terwijl het andere systeem, geïmplementeerd in een Siemens omgeving, de productaanvragen afhandelt voor het vaste netwerk.

Het gebruik van XML aangevuld met asynchrone middleware aan de Enterprise Application Integration kant faciliteert een aantal trends in het bedrijfsleven. Enerzijds kan door de integratiemogelijkheid van diverse systemen, zowel aan de frontoffice als backoffice, de systemen komende uit fusies en overnames gefaciliteerd worden. Anderzijds kan een 24-uurs economie in de frontoffice en batch-windows aan de backoffice ondersteund worden.

Business-2-Business

De problematiek bij business-2-business is soortgelijk aan die van Enterprise Application Integration: grote diversiteit aan platformen, besturingssystemen, netwerkprotocollen en programmeertalen, en het op syntactisch en semantisch niveau koppelen van systemen. De inzet van standaard middleware echter, voldoet niet in een business-2-business situatie. Dit omdat in de meeste gevallen het niet mogelijk is om met de diverse samenwerkende partijen één middleware standaard en één semantische standaard af te spreken.

De inzet van een zogenaamde Integration Broker (IB) biedt de oplossing voor bovenstaande problematiek². Dit omdat één van de functionaliteiten van een IB *transformatie* betreft, dat wil zeggen transformatie op transport- en semantiekniveau (gerelateerd in de XML-stack).

Op transportniveau levert een IB in het algemeen een aantal adapters in diverse categorieën: pakket-, middleware- en netwerkprotocoladapters. Pakketadapters leveren op hoog niveau een koppeling met diverse standaard pakketten, o.a. Siebel, SAP, Baan, Peoplesoft, etc. via de over het algemeen proprietary protocollen en interfaces (API's). Middlewareadapters leveren eveneens op hoog niveau een koppeling aan diverse standaard middlewareimplementatie, o.a. MQSeries, MSMQ, Tuxedo, CICS, HTTP, etc. Tenslotte leveren netwerkprotocoladapters op laag niveau koppelingen met bijvoorbeeld SNA, TCP/IP, etc.

Op semantisch niveau biedt een IB een tweetal mogelijkheden om transformaties te definiëren. Dit kan enerzijds handmatig middels de definitie van transformatieregels en anderzijds automatisch door standaard implementaties van transformatieregels. Hierbij dient niet alleen gedacht te worden aan Biztalk-2-Rosettanet transformaties, maar ook Siebel-2-SAP of Peoplesoft-2-Baan transformaties.

Op syntactisch niveau is XML uitermate geschikt juist doordat hiermee uitwisselbare data gerealiseerd wordt die tussen de verschillende partijen uitgewisseld kan worden.

Bovenstaande technologische invulling van de XML-stack aan de B2B kant van de multi-tier client/server architectuur, stelt bijvoorbeeld een verzekeraar in staat om 24-uur per dag producten te verkopen via een grootgrutter (de frontoffice) terwijl de feitelijke afhandeling van de aanvraag afgehandeld wordt bij de verzekeraar. Dit doordat het verzekeringsbedrijf zijn systemen geïntegreerd heeft met het verzekeringsadviesstelsel van de grootgrutter. Op basis van een offerteaanvraag van de grootgrutter wordt een bericht verstuurd naar de verzekeraar. Deze IB bij de verzekeraar neemt de benodigde transformatie voor zijn rekening.

Het gebruik van XML aangevuld met een Integration Broker faciliteert de trend van (internationale) samenwerkingsverbanden, de zogenaamde business-2-business communicatie.

Conclusie

Als aan de hand van de XML-stack wordt gekeken naar de mogelijke toepassingsgebieden van XML, valt op dat in ieder van de gevallen de inzet van XML niet de problematiek van transport en semantiek, ofwel, het ontbreken van standaardisering van transport, en tags en hun betekenis, oplost. Op het gebied van semantiek is niet zo dat er geen initiatieven op dit gebied zijn. Een eenvoudige zoektocht op het Internet brengt een veelvoud van XML-standaarden aan het licht, variërend van bekende als Rosettanet en Biztalk, tot zeer specifieke als bijvoorbeeld FinXML.

Strikt genomen brengt dit alleen dan een potentieel probleem met zich mee als een organisatie met externe systemen moet communiceren (B2B), of door aanschaf van standaardpakketten of fusies en samenwerkingsverbanden andersoortige XML en transportmechanismen binnen de muren van de IT afdeling komt. Door de inzet van een Integration Broker kan dit probleem relatief eenvoudig worden opgelost.

Auteurs

Drs. P.J. Koning en Drs. B.A. Groenewoud adviseren in hun rol van Technology Consultant bij Cap Gemini klanten uit diverse branches bij de realisatie van een E-Business en/of Enterprise Application Integration strategie middels het gebruik van state-of-the-art technologie.

² De inzet van een IB kan ook in het invulling van de EAI strategie een belangrijke rol spelen. Dit omdat een IB naast transformatie- ook routingfunctionaliteit levert. Het gebruik routingfunctionaliteit vermindert het aantal fysieke koppelingen van $n \times (n-1)$ tot n (waarbij n het aantal applicaties betreft). Het gebruik van een IB voor de invulling van de EAI strategie wordt hier slechts voor de volledigheid genoemd, doch is geen onderdeel van dit artikel.

Literatuur

[1] Het hoe, wat en waarom van applicatieservers – keuze tussen eigentijdse modellen van Microsoft en NOIS-alliantie, Drs. P.J. Koning en Ir. F. Groen, november 1999, Software Release Magazine.