

Service Oriented Architecture bij OAD

BUS NAAR DE DUTCH MOUNTAINS

Al sinds de allereerste dagen van webservices worden reisorganisaties als voorbeeld in presentaties gebruikt om het voordeel van servicegeoriënteerde architecturen aan te tonen. OAD liet het niet bij PowerPoint en voerde een nieuwe IT-structuur in. Een interview met Ivar Scheffelaar Klots (OAD) en Jan-Willem Torken (onafhankelijk consultant).

Door Dré de Man

OAD Reizen B.V. is in de loop van de jaren uitgegroeid van een Overijsselse autobusdienst (1924) tot de op één na grootste reisorganisatie van Nederland. Deze groei, gecombineerd met een nieuwe beleidsvisie vroeg om een nieuwe ICT-omgeving. Ook de eisen die webtechnologie stelde gecombineerd met de eis snel te kunnen reageren op veranderende behoeften van de business, lieten nauwelijks een andere keuze. Het oude systeem was technologisch gezien ook wel erg verouderd. Het was in feite een Basic-variant op een flat file structuur die op een aantal Unix servers draait (Provide X). Uiteindelijk is de keuze gemaakt voor een totaal nieuwe ICT-omgeving.

We praten bewust niet over silo's

De nieuwe ICT-omgeving moest uiteraard genoemde beleidsvisie faciliteren, maar daarnaast uitgaan van een soort zelfbedieningsconcept. OAD werkt als reisorganisatie aan de ene kant voor de klant en aan de andere zijde voor de leverancier.

Scheffelaar Klots: "Touroperators doen heel veel voor leveranciers; het bewaken van voorraden, het sturen van overzicht-

ten; voor airlines is dat een variatie op hetzelfde thema. Al met al zijn dat heel veel activiteiten die wij geautomatiseerd hebben voor leveranciers enerzijds en de consument anderzijds. Het streven is in die nieuwe omgeving om daar ook een zelfbedieningsconcept te introduceren, dat we het werk maximaal willen terugleggen bij die consument respectievelijk leverancier. Wij willen dat wel optimaal technisch met webtechnologie faciliteren, op zodanige wijze dat wij ook de regie van die processen in handen hebben. Daarnaast willen we ook langs die route bereiken dat we de totale fulfillment aan beide zijden beter kunnen stroomlijnen, efficiënter en goedkoper kunnen maken. Dat alles was een belangrijke start voor ons project 'Dutch mountains', in feite de totale nieuwbouw van de ICT-omgeving voor de gehele organisatie."

Modules

Flexibiliteit was een belangrijke eis, en daarmee ook een voorname reden om te kiezen voor een SOA-structuur. Scheffelaar Klots: "Een belangrijke doelstelling is het neerzetten van een omgeving die zo flexibel is, die je zo relatief gemakkelijk kunt aanpassen, zodat je ook veel soepeler kunt inspelen op vragen vanuit de business. Ook dat vereist een bepaalde architectuur. Wat je in zijn algemeenheid ziet bij reserveringssystemen, is dat ze dat ze langs organische groei tot stand zijn gekomen en dat ze doorgaans ook één grote integrale applicatie vormen. We hebben nu ook op voorhand

de principiële keuze gemaakt dat we niet één grote integrale applicatie gaan bouwen waarbij alles uiteindelijk aan elkaar hangt, maar dat we het opdelen in een aantal losse applicaties, modules. Om ervoor te zorgen dat die losse modules wel met elkaar kunnen communiceren hebben we de Solid (Progress) Enterprise Service Bus geïntroduceerd. De SOA-structuur die wij hier hebben opgezet biedt flexibiliteit, autonomie van applicaties en goede optimale integrale functionaliteit.”

De totale functionaliteit werd daartoe verdeeld over een aantal op zich logische modules. Zo is er een module voor het voorraadbeheer, een calculatiemodule, een module factureren, reserveren – in totaal ongeveer 40 modules. Het self service-concept speelt daarbij ook een belangrijke rol. Torken: “Als ik het chargeer, op het moment dat ik vanuit een reserveringsmodule moet zijn in een voorraadmodule, dan weet ik helemaal niet of die voorraadmodule iets is wat wij zelf hier hebben of dat het toevallig aan de andere kant van ‘de plas’ is. Dat is één voorbeeld, zo zijn er legio. Die modules worden zo opgezet dat ze nooit weten of ze nu praten met

een module intern of extern, waar dan ook. Die flexibiliteit is natuurlijk deels theoretisch, je moet toch vormen van transformatie toepassen om elkaar te begrijpen, maar daar hebben we ESB voor.”

Is er bij het ontwerp van de afzonderlijke modules wel rekening gehouden met de andere modules, is er een gezamenlijk datamodel bijvoorbeeld en hoe fijnmazig is het?

Torken: “Je hebt het wel over één van de moeilijkste dingen. We praten bewust niet over silo’s, maar we hebben wel gezegd: we blijven groeperen in applicaties, en daarbinnen hebben we interne en externe functies, of *exposures*, hoe je het ook wil noemen. De crux is natuurlijk wel dat mensen aan de ene kant moeten blijven denken in hun eigen stukje. Je hoeft niet te weten wat er aan de andere kant gebeurt en dat mag je ook niet weten. Aan de andere kant moet je het toch op elkaar afstemmen want het is een beetje zonde om allerlei ouderwetse problemen boven tafel te halen. Als het klantnummer bij de een 8 groot is en numeriek en bij de ander 27 groot en alfanumeriek, dan proberen we dat te vermijden. Het is constant spelen tussen dezelfde dingen



Foto: Dré de Man

Scheffelaar Klots (links) en Torken: “We maken niet alleen een efficiency-slag in de zin van handling, maar ook een grote kostenbesparing”.

Er wordt ook gepubliceerd vanuit het basis-CMS naar gidsproductiesystemen

doen zodat je zo weinig mogelijk hoeft te doen om met elkaar te kunnen blijven praten en iedereen terugzetten in zijn hok en zeggen: en nou zelfstandig blijven denken. Wat betreft de fijnmazigheid: ik vind veertig services wel genoeg. Verder proberen we binnen de 'silo's' aan objectoriëntatie te doen, zodat je hergebruik kunt forceren in andere silo's. Soms zie je ook mensen die grenzen van de silo overtreden, dat is het spel."

In welke zin? Gaan ze direct dingen van andere applicaties aanroepen, buiten de services om?

Torken: "Ja, die neiging blijft altijd bestaan. Inmiddels is het wel zo dat we werken met een framework van Progress en dat dwingt wel een zekere discipline af."

Er is ontwikkeld gebruikmakend van de Progress 4GL; ook bij het begin van het project (2004) geen voor de hand liggende keuze.

Scheffelaar Klots: "Een aantal aspecten in een willekeurige volgorde: ik heb zelf al zeer lang ervaring met Progress, vanaf eind jaren '80, en heb er buitengewoon goede ervaringen mee opgedaan. Een ander belangrijk element bij de totale afweging was de *total cost of ownership*, en de vraag of er voldoende kennis in de markt is van Progress om ook aan ontwikkelcapaciteit te komen. Maar uiteindelijk is de afweging toch een economische, rekening houdende met een aantal technische kwalificaties."

De ESB staat er, doet een aantal transformaties, wat draait er nu?

Torken: "We hebben hier nu twee sporen lopen, aan de e-commerce kant hebben we de meeste meters gemaakt, aan de back-office kant hebben we uiteindelijk gekozen voor een aanpak waarbij alles in één keer live gezet wordt. Het bleek dat het heel moeilijk was om het oude systeem stuk voor stuk te ontmantelen. Aan de randen lukt het wel, betalingsverkeer is op die manier via een webservice geregeld. Factureren, reisdocumenten versturen, aan de voorkant het zoeken en boeken via de websites. Maar bijvoorbeeld bij de voorraadfunctionaliteit waren er te veel afhankelijkheden in het oude systeem."

Ook het CMS is vervangen.

Scheffelaar Klots: "Bij het nieuwe systeem hebben we gekozen voor een basis-CMS, daar vind je alle content terug

uit het bedrijf, alles wat met tekst en plaatjes te maken heeft. Als gebruikers daarmee klaar zijn publiceren ze het naar een web-CMS. Daar zit een aparte zoek-engine bij; die stelt ook eisen aan opslag en dergelijke. Er wordt ook gepubliceerd vanuit het basis-CMS naar gidsproductiesystemen. Onderhoud van dat soort gegevens vindt allemaal centraal plaats, publicatie loopt uiteraard via de bus, want je zou je kunnen voorstellen dat je niet alleen publiceert naar je eigen kanaal-specifieke CMS, maar naar een andere partij. Alles wat over lay-out en opmaak gaat zit in je targetsysteem en niet in je basissysteem, dat kent geen lay-out."

Functiescheiding vindt niet alleen bij het CMS plaats.

Scheffelaar Klots: "We trekken ook op ander plaatsen dingen stevig uit elkaar: functies waar veel rekenwerk aan te pas komt, door de rekenlogica en de data volkomen te separeren. Een aantal van die deelprocessen wordt strikt gescheiden. Dat alles draagt ertoe bij dat je applicaties gemakkelijker te onderhouden zijn. Dat vereist wel de discipline dat je de dingen op de juiste plaats oplost. Je mag geen workarounds bedenken die ogenschijnlijk wel het juiste effect bereiken maar op de verkeerde plaats hun oplossingen met zich meebrengen."

Is er al nu al voordeel behaald door het nieuwe systeem?

Scheffelaar Klots: "Voordeel is er per definitie, alleen het voordeel laat zich op het ene onderdeel gemakkelijker vertalen dan op het andere. Gedrukte brochures vormen een belangrijk communicatiemiddel in de reisbranche. In de oude situatie hadden we een CMS, de data die daaruit kwamen gingen naar een prepress bedrijf en die zorgden dat er mooi opgemaakte pagina's uitkwamen. Dat doen we nu niet meer; de gidsen die wij nu voor de komende winter hebben geproduceerd zijn volledig in het eigen systeem gemaakt. Hier stoppen we de content er nog steeds in, maar de nieuwe omgeving maakt op basis van templates daar automatisch mooi opgemaakte InDesign-pagina's van."

De logica om de templates te vullen zit in een aparte applicatie. We maken niet alleen een efficiency-slag in de zin van *handling*, maar ook een grote kostenbesparing. Je hebt ook een heel ander proces, waardoor je met actuele data kunt werken en aan de andere kant heel vroeg in het jaar al content kunt invoeren, dat staat volkomen los van je opmaakproces. Maar we maken zo'n 14.000 pagina's per jaar.

Het levert een grote besparing op en dat is een grote stap, we hopen er zo nog een paar te maken.”

Performance

Bij SOA denk je al snel aan performance. Is daar niet te veel verlies?

Scheffelaar Klots: “Ik denk dat het per definitie zo is dat wanneer je er iets tussenzet het altijd langzamer gaat. Maar als de gebruiker daar niets van merkt, waar gaat het dan om? Ook dat is overigens een uitdaging, hoe krijg je het zo geconfigureerd dat je alleen de positieve dingen ervan kunt oppakken en dat je nadelen – theoretisch een afnemende performance – niet zodanig zijn dat je er hinder van ondervindt.”

Wordt het te druk dan zetten we er gewoon een servertje bij

De informatie op reissites is vaak niet up-to-date, zoals ik onlangs bij een paar concurrerende organisaties heb mogen ervaren. Hoe zit nu bij het oude en het nieuwe systeem?

Scheffelaar Klots: “We hebben natuurlijk een gigantisch reisaanbod, en dat verandert ook voortdurend, vierentwintig uur per dag. Wij tonen de voorraad meteen op onze website. Nu nog niet helemaal actueel, maar met het nieuwe systeem wel. Zo heb ik grip op de belasting van ons reserveringssysteem: als ik de buitenwereld erop los laat en het wordt op een dag druk omdat het al wekenlang regent en iedereen naar Turkije wil, dan komt er een moment dat je de performance van je reserveringssysteem gaat vernachelen. Daar heb ik daar totaal geen last van. Het is een geclusterde omgeving en wordt het te druk, dan zetten we er gewoon een servertje bij.”

De ontwikkeling van service-georiënteerde modules en applicaties vergt ook een mentaliteitsverandering van de ontwikkelaar.

Scheffelaar Klots: “Aan de ontwikkelzijde is het een heel andere manier van denken. Je moet echt uitleggen dat we dingen uit elkaar hebben getrokken om de toepassing transparanter te maken. Het is wel belangrijker geworden dat je de goede dingen op de goede plaats oplost. Het besef van de voordelen van de nieuwe architectuur dringt ook wel door. Als ik mijn rekenlaag ontkoppeld heb van mijn data laag, dan heb ik ook iets heel anders bereikt, namelijk dat ik een universele rekenstructuur kan maken die voor elk type reis werkt. Of ik nou drie dagen naar Berlijn ga met twee tussenovernachtingen of ik heb een vierweekse rondreis door

Latijns-Amerika met allerlei toeters en bellen, ik werk met één en dezelfde rekenstructuur. De variabelen die ik vanuit een data entry programma aanbied bepalen wat er nu in werkelijkheid voor welke individuele reis doorgerekend wordt. Dat maakt niet alleen de betrouwbaarheid van je calculaties veel beter, maar ook de onderhoudbaarheid. Je moet als ontwikkelaar wel de discipline hebben dat je je er nooit toe laat verleiden variabelen te stoppen in je formules. Maar doe je het goed, dan worden complexe calculaties ineens relatief eenvoudig. Facturering is een goed voorbeeld. Een hele complexe calculatie in deze business is een deel van het yield management. Wij hebben grote garantievoorraden zoals alle grote touroperators. We kopen vliegtuigen vol met stoelen, of wij ze nu verkocht hebben of niet, we moeten ze wel betalen. Het gaat dus om heel grote financiële risico's. Dat betekent dat je wel moet waarnemen waar je een partij onverkochte stoelen dreigt te gaan krijgen. Voor al die problemen geldt dat ze alleen bij een goede scheiding van data, business logica en variabelen goed in een applicatie op te lossen zijn.”

Scheffelaar Klots is zich terdege bewust van het belang van business processen.

“Wat je in zijn algemeenheid bij applicaties in de reisbranche ziet, is dat die bijna altijd zijn gebouwd om de verkoop beter te faciliteren. Het zijn, als ik het wat onaardig vertaal, veredelde factuureersystemen. De handling voordat een reis verkocht wordt, is vaak een meer ondergeschoven gebied. De voorzijde en de webwereld is van belang, maar aan de achterzijde is het meeste te halen ook ten opzichte van nu. Je moet zorgen dat beide goed met elkaar samenwerken en dat betekent ook een herinrichting van je bedrijfsprocessen daarbovenop. Ik ben begonnen met het herdefiniëren van een aantal kernbedrijfsprocessen, die een heel andere verdeling te geven zodat je het optimaal kan inrichten en op basis daarvan je ICT inrichten. Meestal gebeurt het omgekeerd. Dat betekent dus dat de ingebruikname van de nieuwe systemen ook een stuk *change management* inhielden, en dat is een belangrijke hoofdvoorwaarde geworden met de nieuwe applicaties. De applicaties moeten doen wat wij ermee bedoeld hebben, maar de business moet meegaan. Beide moeten gaan werken, als een van de twee het niet doet dan valt het weg. We hebben de beide sporen te lopen en op zichzelf is dat een grote klus, het bouwen van de omgeving is een grote klus waar we drieënhalve jaar voor nodig hebben. Eind volgend jaar moeten we klaar zijn, maar het veranderingsproces in je organisatie dat is een klus van een andere orde. Daarbij moeten mensen ook anders gaan denken en dat is altijd veel lastiger. Als je een applicatie vervangt dan kun je zeggen: die *delete* ik wel even en daar neem ik een andere voor, die machine doet het wel. Maar om mensen zover te krijgen dat is een uitdaging, ook al lukt dat nu heel aardig.”

Dré de Man is hoofdredacteur van Software Release Magazine.