

Webapplicatie van Centraal Boekhuis

Belangrijke rol voor de database

Centraal Boekhuis verzorgt de informatieverstrekking over, en de opslag, distributie en vervoer van boeken en logistiek verwante artikelen. Het heeft een zeer dominante positie in de Nederlandse markt en is voor de helft eigendom van boekhandelaars en voor de andere helft van uitgeverij. Een oude DOS-applicatie voor het bestellen van boeken door winkeliers is recentelijk vervangen door een web-applicatie. Optimize sprak met Erwin Groenendal technisch directeur van Cumquat en ir. Toon Koppelaars, senior IT-architect van Centraal Boekhuis, over dit project.

Wie wel eens in een boekhandel komt, heeft ongetwijfeld wel eens de DOS-applicatie gezien, waarmee boekhandelaars tot voor kort hun boeken bestelden. Per jaar werden daar circa 55 miljoen bestellingen mee doorgegeven. Gezien de steeds slechtere ondersteuning van DOS onder Windows werd het tijd om de DOS-applicatie te vervangen door een modernere applicatie.

Koppelaars: 'De DOS-boxfunctionaliteit gaf continu problemen. We kregen daarover heel veel helpdesk-telefoontjes. Er waren echter ook functionele redenen om de applicatie te vervangen: wij wilden gewoon veel meer informatie aan de boekwinkels gaan geven. De oude applicatie gaf voornamelijk informatie omtrent de vraag welke boeken je kunt bestellen, en bood de mogelijkheid een bestelling te plaatsen. De boekhandel kreeg echter steeds meer behoefte om het hele traject te kunnen volgen, van de bestelling en het moment dat die bij ons inge-

pakt wordt tot aan de levering. Op die manier weet de winkelier precies dat een bepaald boek zaterdagmiddag geleverd wordt en in de vijfde doos zit. We wilden die functionaliteit ook niet meer niet bijbouwen in die oude applicatie.'

Arbeidsintensief

De DOS-applicatie bestond eigenlijk uit twee elementen: een applicatie en een cd-tje waar het hele artikelbestand op stond. Dat cd-tje met alle titels werd elke maand gebrand. Daarnaast bellen de boekhandelaars per dag in om de nieuwe titels en prijswijzigingen op te halen en op te slaan.

Koppelaars: 'Voor het bouwen van de applicatie hadden we twee alternatieven. De eerste was gewoon een goede decentrale applicatie bouwen (fat client) die dan in de boekwinkels zou draaien en via een modem zou communiceren: een lokale database applicatie met Visual C++ bijvoorbeeld. De tweede zou een webapplicatie zijn. Bij een decentrale applicatie moet je die hele gegevensbak in 1.500 winkels neerzetten en dat hele distributiekanaal ervan ook inrichten. Dat zou bij de overgang van DOS naar Windows gehandhaafd blijven, maar het is nogal arbeidsintensief, zowel voor ons als voor de winkelier. Daar wilden we gewoon vanaf. Wat natuurlijk ook meespeelt is onze uitstraling naar de markt: je oogt natuurlijk professioneler wanneer je een webapplicatie maakt.'

Gedurende anderhalve maand heeft men zich bij het Centraal Boekhuis bezig gehouden met de vraag *make or buy*. Er zijn bezoeken geweest aan referentiekanten van een pakket dat in de UK gebouwd is. Daar is uiteindelijk het besluit uit voortgekomen dat het gebouwd moest worden. Daarna is er nog heel even nagedacht over J2EE versus .NET, maar gezien de cultuur van het bedrijf viel de keuze al snel in het voordeel van J2EE uit.

Koppelaars: 'Het is een Oracle-bolwerk hier. We hebben ook voornamelijk AIX in ons serverpark en niet Windows 2000. We gebruiken Windows eigenlijk alleen op de desktop. Wij zijn

"Het is een Oracle-bolwerk hier. We hebben ook voornamelijk AIX in ons serverpark en niet Windows 2000"



Toon Koppelaars, senior IT-architect van Centraal Boekhuis

al sinds 1993 met Oracle bezig. We hebben in de jaren negentig een megaproject uitgevoerd om het mainframe te vervangen, wij gebruiken nu Oracle primair voor onze interne bedrijfsprocessen. Onze gehele logistiek ondersteunen we dus met Oracle. Als je door de hallen loopt, zie je overal schermpjes waar elektronisch getoond wordt, welke boeken gepikt (*klaargemaakt voor verzending, DdM*) moeten worden. Voorheen had men papieren piklijsten. Nu is het te zien op de schermpjes, waarachter een realtime Oracle database zit. In het mainframe tijdperk werden die lijsten 's nachts in batch uitgedraaid, en er werd ontzettend veel papier geproduceerd, nu komt dat op het scherm en wordt alles realtime in de Oracle database geregistreerd. Wij hebben ons eigen ERP-pakket gebouwd in de jaren negentig. Dat is een compleet artikelbeheersysteem, relatiebeheersysteem, orderontvangst, orderprocessing, orderpreprocessing tot en met het reserveren van voorraad in de hal, en de bevoorrading van de diverse magazijnen. Het is allemaal gemaakt met de traditionele tools van Oracle zoals Designer, Developer en Forms. Daarbij hebben we veel gebruik gemaakt van externe partijen uit die tijd.'

J2EE

Toen er eenmaal voor J2EE gekozen, bleven er nog wat vragen open.

Koppelaars: 'Je hebt binnen J2EE nog een hele serie keuzes: welke tools, welke vendor, welke applicatieserver, open source of BEA, Oracle of noem ze maar op. En dan is er ook nog de keuze voor een development omgeving: JDeveloper van Oracle, Eclipse, of JBuilder. Het feit dat we een Oracle bolwerk zijn, heeft ervoor gezorgd dat we honderd procent voor Oracle gekozen hebben: 9iAS en JDeveloper.'

Vervolgens is aan een viertal partijen gevraagd om een voorstel te maken voor de aanpak van een Oracle webapplicatie.

Koppelaars: 'Je krijgt dan vier technologisch en methodologisch verschillende aanpakken. Als je vroeger in de client/server-wereld vier partijen vroeg een Forms-applicatie te bouwen, dan kwamen er vier dezelfde verhalen uit. Ze gingen allemaal met Headstart werken en met CDM. Maar in die webomgeving was dat absoluut niet het geval. Je hebt twee typen partijen op die nieuwe markt: partijen als Cumquat, die gepokt en gemazeld

zijn vanuit de cliënt/server omgeving, de overgang hebben gemaakt naar een J2EE-omgeving maar wel weten waar ze vandaan komen, en je hebt partijen die pur sang Java-bolwerken zijn. Met zo'n partij kun je bij ons geen project doen. Misschien wel bij andere organisaties, maar niet hier.'

Groenendal: 'Het beste voorbeeld vormen business rules. Wij weten dat die in de database horen, dat is de beste plek. Andere partijen zullen die in Java stoppen, want die komen van de andere kant.'

De voorgestelde oplossingen van de andere partijen verschilden nogal.
Koppelaars: 'Er was er een die 100% proprietary tools gebruikte. Dat kon je niet eens J2EE noemen. En dan waren er nog de partijen, voornamelijk de Java bolwerken, die de functie van de

“Er was er één die zelfs met UML aan de slag wilde. Dat is bij ons een verboden woord: UML”

database en wat je daar in zou moeten doen absoluut niet wisten te waarderen of konden plaatsen. Laat ik een stap verder gaan: in de client/server wereld in de jaren '90 zijn we steeds meer gaan werken volgens het principe: als het in de database kán, dan doen we het in de database. Ik had bij het begin van het project al in gedachten dat we het ook in dit project zo zouden aanpakken: de Java-laag zo thin mogelijk, en fat database stored procedures. Wij zullen geen regel Java schrijven als het een regel PL/SQL had kunnen zijn. Die optiek moest matchen en die matchte van die vier alleen bij deze partij. Bij de anderen was er één die een object model wilde maken, en een die zelfs met UML aan de slag wilde. Dat is bij ons een verboden woord: UML.'

Proof of concept

Nadat er voor Cumquat gekozen was, volgden een paar gesprekken waarin de keuze meer in detail werd gemaakt. Groenendal: 'Als je met Java een webapplicatie ontwikkelt, kom je al heel snel op de MVC architectuur, op het MVC design pattern en dan moet je voor alledrie die lagen (m, v en c) een raamwerk kiezen. Daar hebben we op ingespeeld, met BC4J voor de model laag, UIX voor de view-laag, en voor de control-laag hebben we besloten zelf een controller maken. Met die keuzes hebben we bewezen dat we die applicatie kunnen bouwen.'

Koppelaars: 'We hebben eerst een kleine applicatie van vier pages gebouwd om een eerste versie van wat bouwrichtlijnen te produceren, voor dat eerst het project start. Een proof of concept, om het gewoon allemaal even uit te proberen. Zo hebben we in een doorlooptijd van twee maanden eigenlijk de ontwikkelstraat ingericht en als onderdeel daarvan een kleine applicatie: hoe ga je om met development, test en productie, hoe zorgen we dat de software opgeleverd kan worden, het inrichten van de ontwikkelstraat met alle facetten die erbij komen kijken. We hebben toen ook al interne Java workshops gegeven, om de taal en de J2EE-concepten te leren aan het groepje ontwikkelaars dat er toen bij betrokken was.'

Groenendal: 'Het was dus niet alleen de bedoeling dat er een applicatie uitkwam, maar ook dat er een opleiding uitkwam tijdens dat project, zodat zij het beheer van de applicatie kunnen doen. Dat is nu ook zo: het beheer wordt nu door het Centraal Boekhuis zelf gedaan.'

Binnen de ontwikkelafdeling van het Centraal Boekhuis is uiteindelijk webforce-team gevormd van vier à vijf man, dat nu ook na afloop van het project zich op Java en webaspecten zal blijven richten.

Koppelaars: 'Vanwege onze prime directive - zoals ik het altijd noem - 'gij zult geen regel Java schrijven als het een regel PL/SQL had kunnen zijn', hebben we ook bewerkstelligd dat tweederde van de bouw-effort in de database zat en maar een-derde in het Java-gedeelte.'

Bouwfase

Binnen het grote team waren er drie groepen van gelijke grootte, die verantwoordelijk waren voor respectievelijk Java, de database en de interfaces met de eigen backoffice systemen. Groenendal was technisch projectleider van het gehele team.

Groenendal: 'Wij hielden ons vooral met Java bezig, de database werd voornamelijk intern gedaan, interfaces was intern met wat externen, mensen die kennis hadden van de systemen waarmee de koppelingen gerealiseerd moesten worden. Die ontwikkelaars waren dus meer met de database interface bezig dan met Java.'

Koppelaars: 'Nu in de beheersfase is dat eigenlijk nog veel meer omgeslagen. De incidenten die we hebben, lossen we in negen van de tien gevallen op door in PL/SQL iets aan te passen of uit te breiden.'

Het verder verloop van het project, verliep op basis van functioneel prototype.

Groenendal: 'We hebben het proof of concept gedaan en op basis daarvan is besloten om door te gaan. Toen zijn we in eerste instantie functioneel verder gegaan met behulp van een prototype.'
Koppelaars: 'We hebben toen overwogen om volgens de

watervalmethode te werken. Het project stond onder zeer hoge tijdsdruk. Er waren ook steeds meer problemen met de DOS-applicatie. De introductie moest ook op een bepaald moment plaatsvinden, op een voor de boekwinkels rustige tijd van het jaar, en dat is net tussen de piek van de schoolboeken en voor de piek van Kerstmis. Dat werd een uitdaging want in de bouwperiode lagen ook de vakanties. Uiteindelijk is er negen weken gewerkt, en niet eens door iedereen. Er werd besloten om maar twee fases te doen: functioneel ontwerp en bouw. De oude DOS-applicatie deed dienst als functionele afbakening en daaraan waren enkele nieuwe dingen als extra functionaliteit toegevoegd. De bouwfase liep van half juni tot augustus, daarna volgde de eindtestfase, maar er werd ook al getest tijdens het bouwen. Er is opgeleverd in verschillende modules of deelsystemen. De hele applicatie bestaat uit tabbladen, met daaronder de pages voor het functionele cluster. Tijdens het testen bleek dat het in een net iets handiger volgorde gekund had. Van tevoren was al duidelijk wat de wensen van de eindgebruikers waren. Na de bouw hebben we een pilot gedaan in een productie-omgeving met kleine groepjes gebruikers. Daarna zijn we in hoog tempo groepjes van winkeliers gaan aansluiten.'

Storing

Bij het formuleren van de nieuwe functionaliteit waren de eindgebruikers betrokken, net als bij het testen. Hoe zat het uiteindelijk met de tevredenheid van de gebruikers?

Koppelaars: 'In het algemeen was die vrij groot. Toch blijven er dingen die moeilijk in te passen zijn. Zo waren de boekverkopers gewend om met een scanpen grote hoeveelheden boeken te scannen en te bestellen. Dat is in een webapplicatie zeer moeilijk onder te brengen. Er is ook een aantal functies wat trager geworden, maar daar staat een groot enthousiasme over de nieuwe functionaliteit tegenover. Ondanks het feit dat er voor een niet gering aantal winkeliers toch ook een investering in een nieuwe pc tegenover stond, is er vrijwel niemand die nog terug zou willen naar de oude situatie.'

Met de invoer van een webapplicatie nam ook het risico toe op storingen buiten de eigen pc.

Koppelaars: 'Dat viel heel erg mee, we hebben hier sowieso

“Als je kijkt naar waar het interessante Java-deel ligt, dan is dat de controller die we hebben ontwikkeld”



Erwin Groenendal, technisch directeur van Cumquat Information Technology

alles dubbel uitgevoerd. Als er nu een storing is, hebben alle 1.500 boekwinkels tegelijk een storing. We hebben drie grote storingen gehad: op een zaterdag is er een storing geweest doordat een vitale KPN-lijn naar het internet uitviel. Toen hebben we een uur vastgelegen. Daarnaast hebben we twee keer softwarecrashes van het Java-deel gehad. Op zich ook niet erg: je herstart en dan doet hij het weer. Wat wel ingewikkelder is dan bij PL/SQL, is dat je code van Cumquat hebt, maar ook van Oracle en dan ook nog eens de JVM van IBM. In welke code zit dan de fout? Dan val je terug op de leveranciers. Het eerste wat die zeggen is: doe een upgrade naar de laatste patch. Sindsdien hebben we geen crash meer gehad.'

Hoe ging de samenwerking tussen de Java-groep en de PL/SQL-groep?

Koppelaars: 'De planning was zo dat de mensen onafhankelijk van elkaar konden werken. We zaten dicht bij elkaar, zodat je korte communicatie lijnen had.'

Groenendal: 'Het kwam gelukkig zelden voor dat men probeerde de verantwoordelijkheden af te schuiven, bijvoorbeeld javanisten die zeiden dat het aan de database lag of omgekeerd. Fouten die ontdekt werden, bleken ook zelden in de Java-laag te zitten. Dat is fijn, omdat de tijd om iets te fixen in de Java-laag al snel langer is.'

Apentruccje

Het security-aspect is bij deze applicatie ook nauw verweven met de verdeling van de functionaliteit tussen de Java-groep en de PL/SQL-groep.

Groenendal: 'Toon is eigenlijk de architect van het security-deel geweest en dat heeft zich heel goed bewezen.'

Koppelaars: 'De mensen in de PL/SQL groep deden eigenlijk gewoon de views voor de web-pages, de stored procedures en alles wat daarachter zit. Je kunt een view maken die bestelregels toont, maar daar moet de prijsinformatie bij, de titel, de auteur, in combinatie met het security-aspect dat iedere boekwinkel alleen zijn eigen bestelregels mag zien. Bij de DOS-applicatie had je dat niet, want de bestelregels stonden allemaal op je eigen harde schijf staan. Nu staan ze allemaal shared in een grote besteldata tabel. Qua datamodel heb je dan niet een tabel per boekwinkel, maar één grote tabel met alle boekwinkels. Allereerst wilden we het liefst dat de row-level security die rondom zo'n tabel geïmplementeerd moet worden ook niet in Java, maar in PL/SQL terecht kwam. Vervolgens dan in PL/SQL het liefst gecentreerd op één plaats, en niet overal in view en stored procedure code. De VPD-optie (Virtual Private Database) van Oracle bood hiervoor een prima oplossing: alle row-level security programmeer je dan slechts eenmaal uit in wat een policy function genoemd wordt. Dit was voor ons allemaal nieuwe technologie, nieuwe tools, een nieuwe manier van werken. De hele prime directive (*zoveel mogelijk in de database, DdM*) had één grote onderliggende reden en dat was het minimaliseren van het risico. Er zat al zoveel nieuw spul in het project dat we het risico zo miniem mogelijk wilden houden, ook gezien de tijdsduur. We hebben uiteindelijk een applicatie gebouwd die voor javanisten niet zo interessant is.'

Groenendal: 'Nja ...'.

Koppelaars: 'We hebben in ieder geval helemaal niet object-oriented zitten denken, we hebben geen class bestelregels.'

Groenendal: 'Dat niet, maar als je kijkt naar waar het interessante Java-deel ligt, dan is dat de controller (de C van MVC) die we hebben ontwikkeld. Daar maken we wel gebruik van OO-aspecten. Degene die het raamwerk gebruikt is op een heel andere manier met Java bezig.'

Koppelaars: 'Ja, dat is waar. Het werd op een gegeven moment zelfs bijna een apentruccje. Als een van onze mensen een nieuw scherm moet maken, dan pakt hij gewoon een UIX source die er het meeste op lijkt, je past die een beetje aan, je zorgt dat de database view er is en je knoopt dat gewoon aan elkaar met tien regels Java.'

Groenendal: 'Het framework is ontwikkeld om het beheer zo gemakkelijk mogelijk te maken.'

Koppelaars: 'Het raakvlak tussen Java en de database werd ingevuld door middel van het "één-view-per-page" principe. Aan elke html-page werd een view gekoppeld, wiens taak het

was alle gegevens op te hoesten die op die page getoond werden. Al snel krijg je dan views die je niet meer kunt updaten. Zo'n complexe view daar mag je geen update op doen, alleen selecteren. Dat gaf dus problemen bij pages waar ook ge-transact kon worden (bestelregels en retouren): door 'instead of' triggers te gebruiken kun je dit soort views weer updatable maken. Dat had ook weer als bijkomend voordeel dat automatisch alle transactie code in PL/SQL (de 'instead of' trigger) terecht kwam.'

'Wij zullen geen regel Java schrijven als het een regel PL/SQL had kunnen zijn'

Groenendal: 'Er wordt veel gezocht en gelezen en het invoeren betreft voornamelijk bestellingen en retouren. De meest gebruikte functie is het zoeken binnen de applicatie. Dat heeft ook veel extra aandacht gekregen, omdat het een enorm bestand is en de performance daarvan erg belangrijk is.'

Koppelaars: 'Op de database zijn we toch behoorlijk wat aan het SQL-tunen gegaan. Je moet bijvoorbeeld de top tien van SQL-statements die de meeste i/o of cpu nodig hebben gaan monitoren. Met SQL-statement rewrites, hints of andere indexen valt er dan toch nog vaak heel wat te winnen.'

Toekomst

Capaciteitsproblemen zijn er echter nooit geweest, noch met de oude, noch met de nieuwe applicatie. Centraal Boekhuis heeft bovendien gekozen voor grote machines, en kan processor capaciteit eenvoudig naar behoefte toevoegen. Nu de web-applicatie goed blijkt te werken, wordt het tijd om aan de toekomst te denken. Toekomstige uitbreidingen zullen te vinden zijn in het ontwikkelen van webservices, bijvoorbeeld eentje die zoekfunctionaliteit op sleutelwoord aanbiedt. Daarvan zouden boekhandels met een eigen website gebruik van kunnen maken. Centraal Boekhuis verwacht in 2004 diverse van dit soort webservices te kunnen gaan bouwen.

Tekst en fotografie: Dré de Man