Datamining voor de massa's

Datamining naar Oracle-kernel

Jacek Myczkowski is Vice President Datamining bij Oracle. Hij is afkomstig van Thinking Machines, een bedrijf dat door Oracle vijf jaar geleden opgekocht is in de hoop hun product snel te integreren in het Oracle productenaanbod. Dat pakte anders uit: uiteindelijk kwam het terecht in de database-kernel en vond het een bredere toepassing. De mogelijkheden voor datamining zijn er alleen maar op vooruit gegaan.

Datamining lijkt me een beetje een misleidende term: het klinkt alsof je naar data moet graven. Gaat het niet eigenlijk om data die al beschikbaar zijn?

Myczkowski: 'Bij de formele definitie van datamining gaat het om het vinden van verborgen patronen in data. Je kunt het ook vergelijken met technieken die mensen kennen, zoals query reporting en OLAP. Daarbij weet je welke vraag je wilt stellen, en je probeert een antwoord op die vraag te vinden. Bij datamining weet je de vraag niet waarvoor je het antwoord zoekt. Mathematisch gesproken zijn OLAP en query reporting deductieve technieken, wat betekent dat je dat je kijkt naar het verleden en dat de data gebruikt worden om een hypothese te bevestigen. Datamining daarentegen is een inductieve techniek, wat betekent dat het probeert iets te projecteren op de toekomst, waarvoor de data gebruikt worden om genoeg significantie te verzamelen om te zeggen dat het waarschijnlijk zal gebeuren.

Voor mij is dat de beste definitie vanuit het gebruiksperspectief: het gaat om wat er gebeurt versus wat waarschijnlijk zal gebeuren. De vragen over het verleden zijn interessant als leer-ervaring, maar als je iets kunt afleiden heb je een mogelijkheid om datgene wat waarschijnlijk gaat gebeuren te beïnvloeden. Als je kunt voorspellen wie er waarschijnlijk van je gaat kopen, wie er waarschijnlijk als afnemer van je diensten weg zal gaan, heb je nog steeds een mogelijkheid om erop te reageren. Het probleem is echter dat zoiets gebaseerd is op statistische gegevens en statistieken zijn niet altijd waar, ze bezitten maar een zekere waarschijnlijkheidsgraad. Hoe beter je er in slaagt die waarschijnlijkheden te vinden en te definiëren, des te waarschijnlijker wordt het dat die gebeurtenissen ook zullen plaatsvinden. Dat is het gebied waar het bij datamining om gaat. Historisch gezien ontstond het zo'n vijftien tot twintig jaar geleden, en het heeft een nieuwe stimulans gekregen door de komst van wat ik *powerful computing* zou willen noemen. Er is behoorlijke rekenkracht voor nodig.'

Esoterisch

Myczkowski: 'Voor Oracle werkte ik bij Thinking Machines, een bedrijf dat massieve parallelle computers bouwde. Daarnaast bouwden we applicaties, die verband hielden met defensie en heel grote wetenschappelijke toepassingen, business computation van technische aard, zoals olie-exploitatie. We zochten in de commerciële wereld naar toepassingen die zo'n grote rekenkracht nodig hadden. Toen heette het nog niet datamining maar knowledge discovery, en wij probeerden deze technieken te gebruiken om gedragingen af te leiden. Onze eerste klant was American Express. We hebben een systeem voor hen gebouwd dat de waarschijnlijkheid van in gebreke blijven bij betalingen zou voorspellen. Daarmee zouden ze pro-actief kunnen optreden om verlies te beperken. Dat was het eerste systeem van die aard, en werd ook gedeployed bij American Express onder de naam Quantum. Dat is het waar het bij datamining om gaat. Het feit dat rekenkracht goedkoper is geworden, heeft datamining tot een minder esoterisch gebied gemaakt, iets wat het gewone bedrijfsleven nu zou kunnen toepassen. De perceptie is dat het alleen maar voor een kleine groep weggelegd is omdat er zeer geavanceerde skills nodig zijn voor het toepassen van deze technieken, maar ook dat is aan het veranderen. Ik denk dat je steeds meer zult gaan zien wat ik "datamining voor de massa's" zou willen noemen. Wanneer de techniek succesvol zal zijn in de markt, zul je een manier moeten vinden om het een stuk gemakkelijker in het gebruik te maken, en om het te populariseren over vele applicaties en toepassingen. Het is er nog niet, maar ik denk dat het de potentie heeft om die populariteit te bereiken.'

Klikken

Oracle Datamining heeft een dashboard dat de indruk wekte dat het zelfs voor de business gemakkelijk te gebruiken zou zijn. Dat lijkt enigszins bedrieglijk beeld.

Myczkowski: 'Oracle richt zich op een heel breed spectrum van gebruikers. Ik werk aan de technologie-kant: de server en de database. Daarom komen we mensen tegen die heel 'sophisticated' zijn op het gebied van data-analyse en mensen die dat juist niet zien. We proberen beide soorten gebruikers van dienst te zijn. Je hebt *automation* en *ease of use*. *Automation* betekent meestal dat je een paar opties neemt die passen bij die situatie of dataset zoals de ontwerper van de *automation* dat voorzien heeft. *Ease of use* betekent dat je precies weet wat je wilt en dat je de details ervan wilt controleren. We willen ons op beide uitersten richten. Een zeer gevorderde gebruiker kan voordeel hebben van een interface die eenvoudig te gebruiken is. Datamining voor de massa's is waarschijnlijk veel meer gebaseerd op *automation*, datamining voor de technische specialist zal veel meer op *ease of use* zijn gebaseerd. We proberen beide te doen, in ons product is er een optie die het mogelijk maakt iets met een enkele klik te doen. Je wijst bijvoorbeeld naar de dataset en je zegt: ik wil de waarde van dit veld voorspellen. Dat is alles wat je hoeft te doen, al het andere is geautomatiseerd. Aan de andere kant kun je dezelfde stap zetten met heel veel klikken en met keuzes voor instellingen, datatransformatie, de waarde van zekere parameters of algoritme-typen die je wilt gebruiken.'

> ***Verplaats niet je data naar het algoritme, maar je algoritme naar de data***

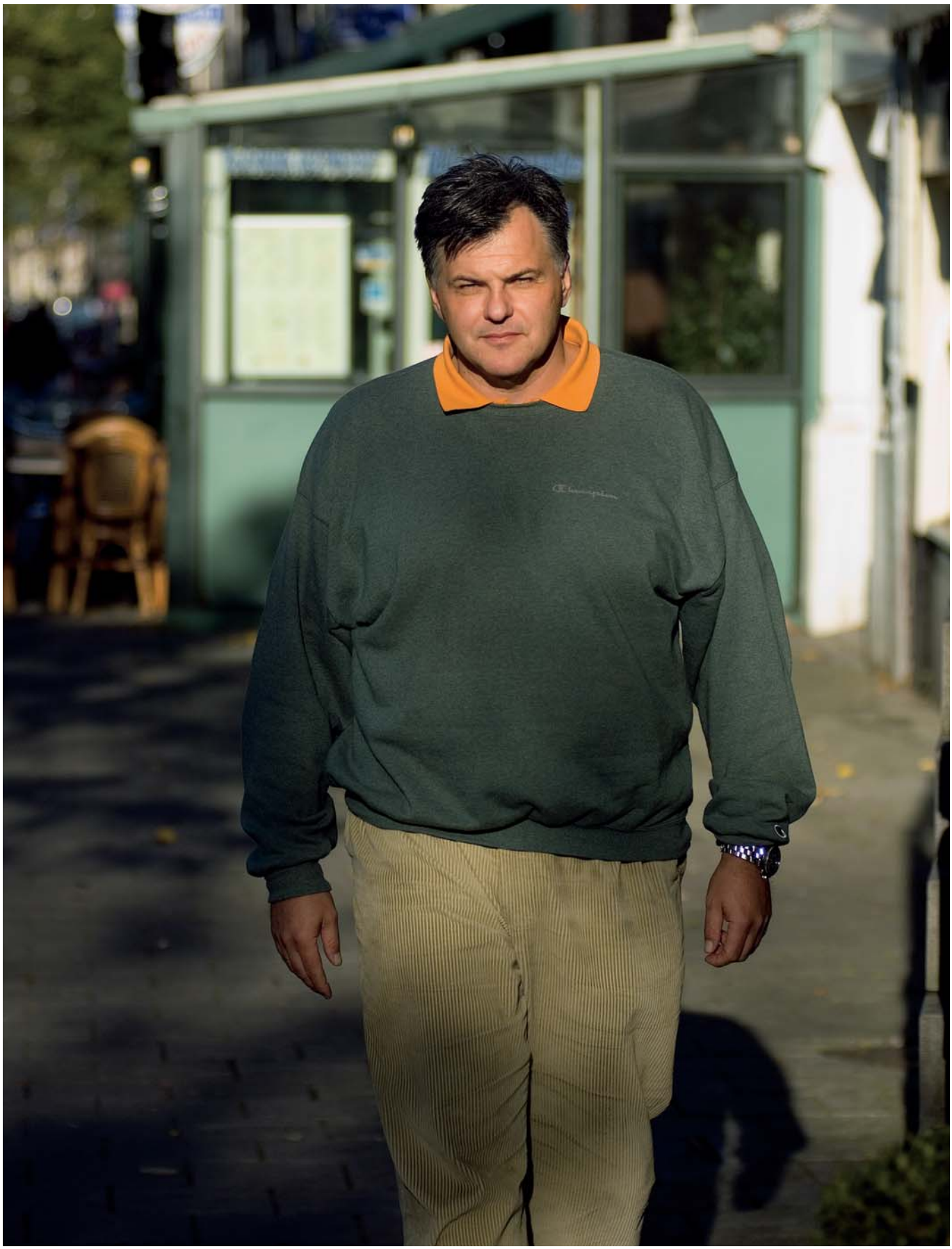
U bent langere tijd werkzaam geweest op dit gebied, dus u zult zelf wel een behoorlijk aantal algoritmen ontwikkeld hebben. Of zit de kwaliteit van dit soort werk niet in de algoritmen?

Myczkowski: 'Eerlijk gezegd is het een tijd geleden dat ik echt code geschreven heb, maar in het verleden deed ik dat wel en schreef ik allerlei soorten algoritmen. Het is waar dat algoritmen de component van datamining zijn waar alles om draait, maar je moet sommige dingen ook met een korreltje zout nemen. Je moet er zeker van zijn dat wanneer je innoveert dat je ook een set van acceptabele normen volgt. Wanneer je het beste algoritme ter wereld verzint, maar niemand gebruikt het behalve jij, dan zul je zien dat men aarzelt om het te gebruiken omdat het gezien wordt als iets wat niet in de praktijk getoetst is. Een betere benadering is dat er een geaccepteerde groep van algoritmen is. Op mijn gebied heb je een redelijk grote verzameling patenten. Je krijgt alleen geen patenten op het algoritme zelf, dat wordt gewoonlijk gepubliceerd in technisch-wetenschappelijke literatuur. Je krijgt een patent op de feitelijke implementatie van het algoritme op jouw platform, dat is wat je doorgaans differentieert. Bijvoorbeeld: bedrijf A heeft iets wat je een *decision tree* noemt, een basaal en erg bekend algoritme op het gebied van datamining, wij hebben er ook een. Wij hebben daar drie of vier patenten op. Niet op het concept van een *decision tree*, maar op hoe je zoiets in de database implementeert. Voor mij is dat het grote avontuur bij Oracle, het conceptuele.

Oracle heeft jarenlang databases gebouwd, het zijn populaire producten, maar databases zijn voor een groot deel archiveersystemen voor je data, die systematiseren hoe je je data opslaat. Het is aardig dat je je data kunt opslaan, maar het zou veel aardiger zijn wanneer je de database kunt veranderen in een *knowledge base*, dat je dat je niet alleen kunt opslaan, maar dat je er bruikbare informatie uit kunt halen, of zelfs kennis die je niet had. De klassieke manier gaat ervan uit dat je opslaat, data extraheert, naar speciale servers verplaatst die analyse uitvoeren of datamining of OLAP, en je resultaten weer terug plaatst in de database om die verder te verwerken of beschikbaar te maken voor andere applicaties. Onze filosofie is: wanneer je die data al in de database hebt, zou je al die dingen daar moeten kunnen doen. Het motto is min of meer: verplaats niet je data naar het algoritme, maar je algoritme naar de data. Vooral nu de hoeveelheid data zo explosief groeit, lijkt het simpeler om de algoritmen die compacter zijn naar de database te brengen dan die enorme hoeveelheden data te verplaatsen iedere keer wanneer je er iets mee wilt doen.'

Avontuur

Myczkowski: 'Ons intellectuele eigendom, wat uniek is aan Oracle's aanpak van datamining, is: hoe neem je technologie als high-end analyse? Dat is niet alleen datamining, maar ook statistiek of features voor biowetenschappen, bijvoorbeeld een algoritme voor het in kaart brengen van genen (BLAST). Hoe embed je dat in de database zodat hij in staat zal zijn dat soort dingen te doen? Dat is voor ons het belangrijkste intellectuele avontuur. Ten eerste heeft niemand dat nog gedaan en ten tweede zijn databases niet gebouwd als analytische machines. Ze zijn erg goed in het opslaan van data, het terughalen in een gestructureerde vorm, maar wanneer je ze vraagt te rekenen dan is dat en stuk moeilijker. We zijn de laatste zes jaar bezig geweest in de kernel van de database een infrastructuur te bouwen die analyse kan doen. Daar zijn zo'n veertig patenten uit voortgevloeid. Als groep zijn we daar echt trots op, want wat we proberen te doen, een database-engine als een analytische engine te laten werken is niet zomaar iets.'

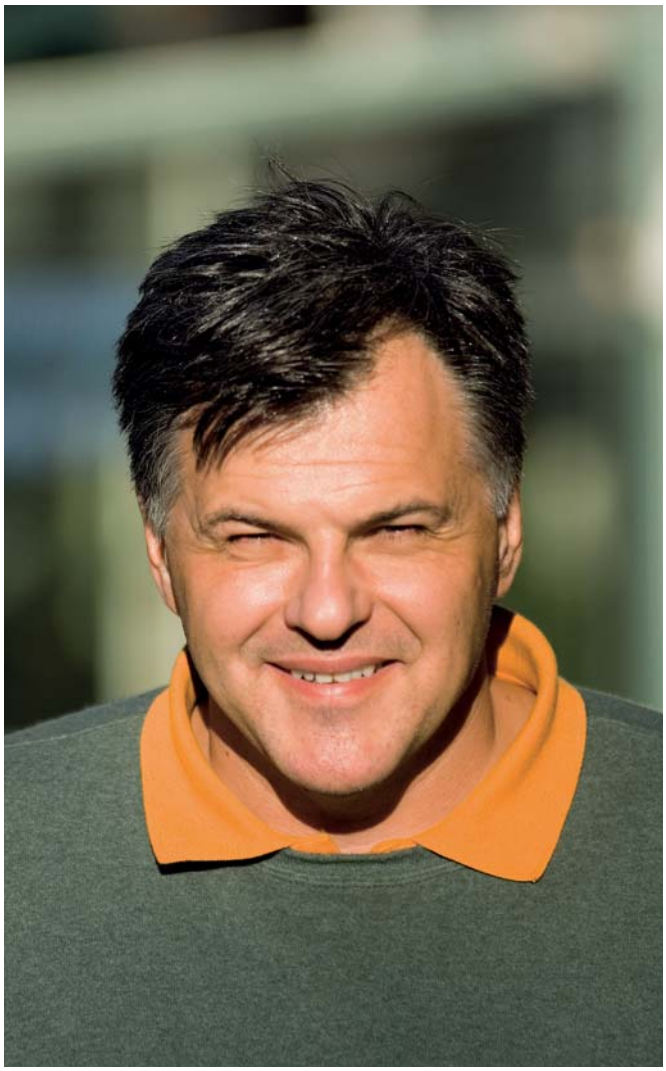


Het idee om de datamining-functionaliteit in de database-kernel onder te brengen is in de praktijk ontstaan.
Myczkowski: 'Thinking Machine werd in 1999 door Oracle overgenomen. Wij hadden een zeer schaalbaar datamining product dat was ontworpen om met heel grote problemen te werken. De schaalbaarheid van het product was vooral wat Oracle aantrok. Het oorspronkelijke doel was dan ook om het product deel uit te laten maken van de Oracle producten, maar het bleek iets avontuurlijker te zijn. Het is een veel langere weg gebleken maar de voordelen zijn - nu terugkijkend op die zes jaar – dan ook veel groter dan we eerder gedacht hadden.

Onze eerste gedachte na de overname was: stop het product in de database. We kwamen er al snel achter dat dat niet zou gaan werken. Die database en het paradigma dat wij gebruikten waren met elkaar in tegenspraak. Na enig nadenken besloten we iets anders te doen, en de database als startpunt te nemen, en ons dan pas bezig te houden met de vraag hoe we al die algoritmen moesten ontwikkelen of gebruiken om in die omgeving te werken. Wanneer je denkt aan een metriek, de kortste verbinding tussen twee punten, denkt iedereen aan Cartesiaanse metriek, een rechte lijn. Als je naar New York City gaat en Cartesiaanse metriek wilt toepassen, zul je echt in de problemen komen. Er is zelfs zoiets als Manhattan-metriek, dat is een grid en vormt de manier waarop je moet handelen in die omgeving. Dat was ook de filosofie die wij hanteerden om in de Oracle-database te kunnen opereren. In plaats van iets op te lossen dat voor de hand liggend was, probeerden we het probleem om te draaien, en zeiden: dit is onze metriek, dit hebben we tot onze beschikking en we gaan dit veranderen. De laatste zes jaar hebben we veel werk verzet om dat te bereiken. Sommige van die algoritmen zijn erg standaard. Het heeft ons bijvoorbeeld behoorlijk wat tijd gekost om de decision tree in de database te implementeren.'

Waarom?
Myczkowski: 'Decision trees maken veel gebruik van recursie. Ze vragen een hoop gebalanceer van unieke taken, want de database houdt ervan om dezelfde belasting te hebben over alle processing units. Het kostte een hoop werk om daar op uit te komen. We losten het in feite op door een aantal nieuwe mogelijkheden in de database te embedden om iets wat daarvoor al bestond op een nieuwe manier te gebruiken. Dat betekent; we probeerden met bepaalde dingen uit te drukken wat er in het product moest gebeuren.'

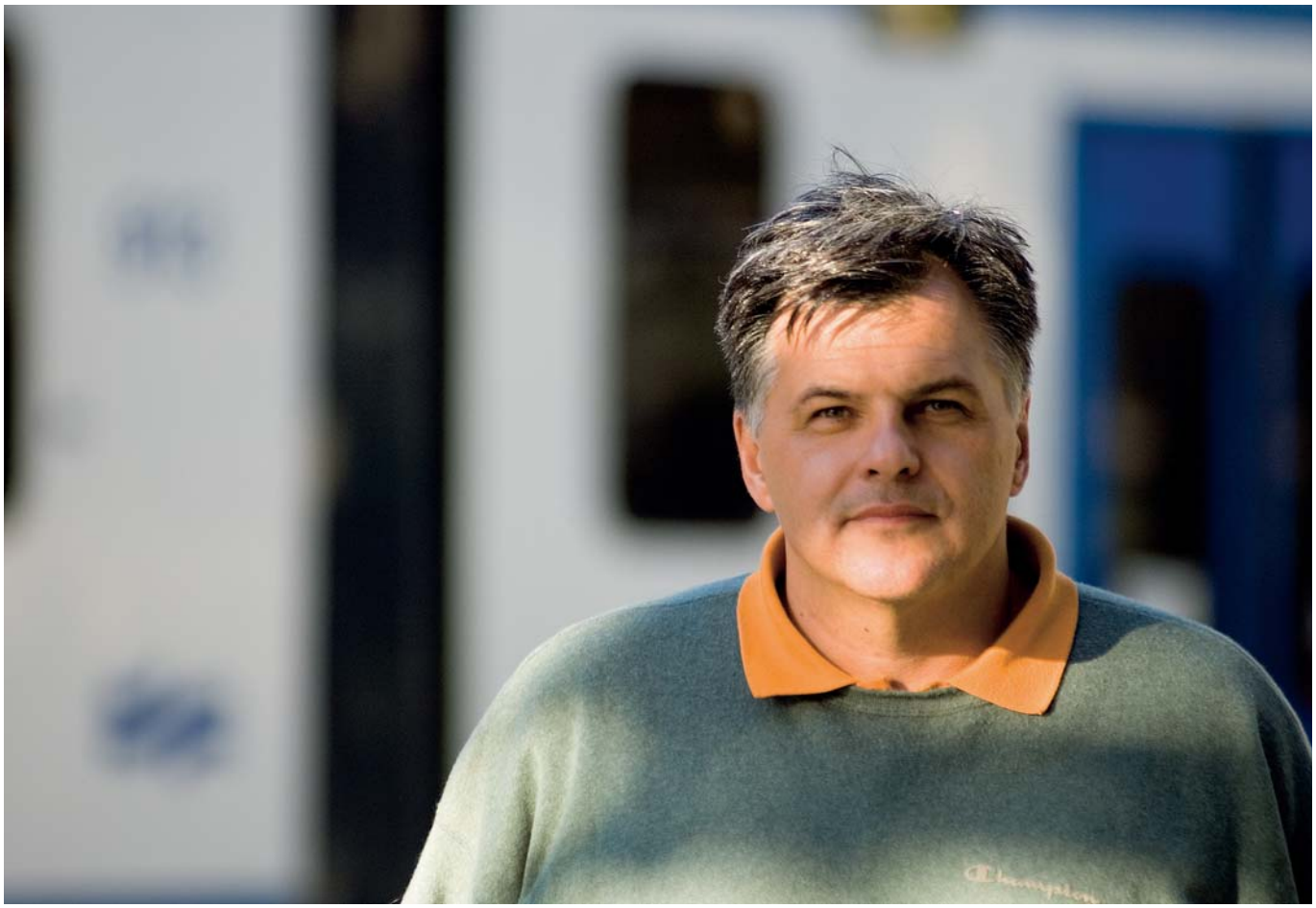
Welke dingen waren dat precies? Ik kan me voorstellen dat een relationele database niet het ideale gereedschap is om …
Myczkowski: 'Veel had van doen met geheugenbeheer, met de vraag hoe je processen start die gemodificeerd moesten worden.'



U bleef waarschijnlijk niet in het relationele model.
Myczkowski: 'Nee, wij bleven binnen dat model, dit is een relationele database en dat was onze *constraint*.'

Anders was u bij uw server gebleven.
Myczkowski: 'Ja, dan hadden we een *stand alone* product gebouwd. Als je een eigen server bouwt, dan bepaal jij je metriek.'

Dat is wat anderen gedaan hebben?
Myczkowski: 'Ja, er zijn twee soorten producten. Veel producten definiëren een server en een databron. Maakt niet uit waar ze vandaan komen, van een flat file, van een database. Het punt is dat het voor hen een databron is en de manier waarop ze het gebruiken is: ze nemen de database, brengen ze in een formaat dat van toepassing is, stoppen het in de server en verwerken het. Als je dat doet, heb een hoop vrijheid, je kunt je dataformaat definiëren, je rekenstijl en je technieken. In ons geval worden data gepresenteerd als een relationele tabel, en niet alleen



dat: je wilt ook dat alles wat daarna komt ook relationeel is. Het concept van mining brengt de issue van een model met zich mee, wij moesten ons aan de database conformeren om een model op te slaan. We moesten het dus opslaan in een formaat dat kon bestaan in de database, dus in een tabel. Tegenwoordig is het een XML-representatie, maar daaronder is het nog steeds opgeslagen in een tabel. Onze aanpak was de relationele wereld meester te worden en onze aanpak te veranderen om die in die wereld te laten passen. Dat is denk ik onze grootste prestatie, omdat de meeste concurrenten compromissen gesloten hebben op de weg daar naartoe. Of ze hebben – hoewel het product geünificeerd is –twee servers die met elkaar communiceren en data heen en weer sturen. Wij probeerden die functionaliteit binnen de kernel te implementeren. Het zit onder de SQL-lijn.'

Geshockeerd

Myczkowski: 'Het was veel werk maar als het eenmaal gelukt is, dan worden er ineens een hoop dingen mogelijk. De database heeft enorme mogelijkheden, je vindt er allerlei soorten manieren om je data voor te bereiden, je vindt een hoop mogelijkheden om dat wat je doet in datamining met andere stappen te verbinden, bijvoorbeeld het bouwen van query's te beginnen. We bouwden een *prediction operator* in SQL. Als je dat doet kun je die query's die voorspellend van aard zijn met standaard SQL combineren. Je kunt zeggen: ik wil alle klanten zien die met een waarschijnlijkheid van tachtig procent niet aan hun betalingsverplichtingen zullen voldoen binnen die postcode of die dit product gekocht hebben tussen april en december. Als je database dit ondersteunt, is het één enkele query, dat geeft veel performance en eenvoud. Het opent veel nieuwe mogelijkheden waar mensen nu pas aan toe komen, want het duurt even voordat de perceptie van een product verandert. Oracle databases worden nog voornamelijk gezien als een storage voorziening. Wanneer we ze vertellen: "Hé, je moet dit gaan zien als veel meer, want er zijn zoveel mogelijkheden," dan zijn ze geshockeerd. Dan beginnen ze zich af te vragen hoe ze daar gebruik van kunnen maken, want uiteindelijk is het voor veel gebruikers een kwestie van kosten. Als ik iets gebruik wat al geïnstalleerd is en deel uitmaakt van mijn infrastructuur, zeker onder een CIO die zich verplicht ziet kosten te besparen, dan is er een flinke druk om er gebruik van te maken. Bovendien heb je nu het voordeel van de extreme schaalbaarheid en dat staat het je toe dingen te doen die je anders niet zou doen.'

Baskets

Zijn er andere delen van de database of programma's die nu profiteren van de features die vanwege de datamining aan de kernel zijn toegevoegd?

Myczkowski: 'Ja, veel van de infrastructuur die wij gecreëerd hebben is aan de 'gewone' gebruikers van de database vrijgegeven. Eén van de algoritmen bijvoorbeeld dat ook populair is in datamining is *association market basket analysis*, welke producten passen bij elkaar. Dit is geïmplementeerd via de techniek *frequent items*, je berekent dingen die bij elkaar passen en die frequent voorkomen. Maar wanneer je erover nadenkt is het alleen een additionele laag op het tellen van verschillende baskets, een basale technologie van tellen, die je min of meer gemasseerd hebt om die associatierollen te krijgen. Veel mensen zijn geïnteresseerd in de notie van het vinden welke items er in je database zitten. Het heet een algoritmische techniek van *frequent item sets* en het is een basis database-operatie die nu deel uitmaakt van de database. Je hoeft niet naar associatierollen te gaan, je kunt het gebruiken. Veel mensen doen dat al voor reporting en onderhoudsdoeleinden. In versie 10.2 van de database zijn nu veel van de dingen die in *decision trees* gebruikt zijn beschikbaar als *generic building blocks*, bijvoorbeeld met betrekking tot regressie. Het doel is: wanneer het meer algemeen van aard is en meer dan datamining dient, maak het dan beschikbaar. De standaardtaal van de database is SQL, dus je moet het in SQL beschikbaar maken. Dat betekent meestal wel enig werk, want je moet een syntax maken, maar meestal wordt het beschikbaar gesteld en dan wordt het een deel van de trukendoos. Je zult in de SQL-manual dan ook nieuwe commands vinden die je dit soort dingen laten doen.' *(zie ook het artikel 'Oracle Analytische Functies' op pag. 7, red.)*

Veel nieuwe producten van Oracle lijken ook goed samen te werken met datamining en wat er bij hoort. Zo kocht Oracle Times10 en in-memory calculaties lijken me voor dit soort dingen heel geschikt.

Myczkowski: 'Dat is een heel recente acquisitie en in het huidige product is niet veel overeenkomst tussen Times10 en wat wij doen. We werken er wel aan om die technologie op een lijn te brengen. Hun technologie moet nu ook binnen de Oracle-paradigma's passen. Ze hadden een enorme cache bijvoorbeeld. Waar gaat dat nu naar toe, als deel van server? Maar veel van de algoritmen die we gebruiken zullen voordeel hebben van de toegang tot zeer grote geheugenruimtes, dat is duidelijk.'

Oracle heeft nu ook een aantal bedrijven gekocht als PeopleSoft, die hun eigen datamodellen gebruiken. Maakt dat uw werk niet erg gecompliceerd?

Myczkowski: 'Zij gebruiken hun eigen modellen en schema's, maar de meeste van die programma's – PeopleSoft of Siebel – hebben veel van hun infrastructuur op Oracle gebaseerd. Oracle was een van de, zo niet het belangrijkste platform. Zolang het opgeslagen is in Oracle kan het ons niet zo veel schelen hoe dat is opgeslagen, want het zit in de Oracle-structuur. Als ik een PeopleSoft schema neem en ik weet wat het is en waar het zich bevindt heb ik toegang tot die data.'

JSR 73

Oracle is spec-lead voor JSR 73, de Java specificatie request met betrekking tot datamining. Deze JSR maakt interoperabiliteit tussen verschillende datamining systemen mogelijk. Ontwikkelaars kunnen ook code schrijven die niet gebonden is aan een datamining-applicatie. De API is geschikt voor zowel gevorderden als beginners en omvat een framework om hem uit te breiden. De ontwikkeling ervan begon in juli 2000 en de final release werd gepubliceerd in augustus 2004.

Myczkowski: 'Thinking Machines had een Java product. Alle user side producten waren geschreven in Java en de server-side producten in C++. Toen we naar Oracle gingen in 1999, was Oracle's verhaal over Java heel erg database-centrisch, er zou veel Java in de database terecht komen. Uiteindelijk is Java meer in de middle tier terecht gekomen, er is een overblijfsel van Java in de database maar veel minder dan het was. Wij voelden altijd dat de toekomst van datamining echt belangrijk zal zijn voor Oracle apps. Oracle heeft er altijd aan vastgehouden dat Java en JEE het platform zouden zijn voor deze applicaties. We besloten dat het aardig zou zijn, wanneer we iets zouden doen om een groter accent op Java te leggen. Er was één persoon in mijn groep – alle eer gaat naar hem – die de JSR startte. Het idee was om met anderen een gezamenlijk gebied te vinden om de gezamenlijke concepten over datamining in onder te brengen. Het is een langzaam proces van het idee naar een officiële JSR en het heeft vijf jaar geduurd. Oracle is de spec lead maar alle belangrijke spelers (SAS, SAP, IBM en vele anderen met als enige uitzondering Microsoft) hebben meegewerkt. In onze 10.2 versie hebben we in feite een API die we JSR 73 compliant noemen. We noemen het JDM, Java Data Mining. We hebben de Java API en de PL/SQL API interoperabel gemaakt: je kunt iets in Java doen en het in PL/SQL doen en omgekeerd. JDM probeert zich te richten op de JEE-gemeenschap en mensen die daar applicaties bouwen. We hebben ook een partnership met SAP, ze bieden een optie aan om wanneer je draait op SAP Business Warehouse, Oracle datamining te gebruiken; de integratie is uitgevoerd via Java. Het voordeel voor iemand die veel in SAP doet is dat het dan beschikbaar is voor de hele SAP stack, dat is zeker relevant in West-Europa. Het is grappig dat we hier zo goed samenwerken, waar we op andere punten in een felle concurrentiestrijd verwikkeld zijn.'

Maar dan moet je die schema's heel goed kennen.
Myczkowski: 'O ja, maar je werkt met de applicatie-groep en specifieke applicaties en misschien heb je een CRM-applicatie die iets doet als marketing intelligence. De mensen die deze applicatie bouwen begrijpen heel goed waar de data zijn en hoe die in elkaar zitten, onze job zal zijn hun technologie ter beschikking te stellen en indien nodig zelfs de juiste modellen daaruit te creëren. Zij zullen ons vertellen welk business probleem ze willen oplossen en welke data beschikbaar zijn. We zullen dan een prototype bouwen en kunnen voorspellende modellen bouwen, en dus zullen de modellen kunnen voorspellen welke data ontbreken, bijvoorbeeld demografische data. Wij zullen ze dan vertellen: jij moet je scherm verrijken met demografische data, of we hebben juist meer dan genoeg informatie maar willen het stroomlijnen.'

Machinaal leren

Als je datamining doet lijkt event processing zoals Progress dat doet ook een interessante optie. Daar wordt min of meer real time data geanalyseerd waarbij gebruik gemaakt wordt van redundantie: alleen die data die van belang zijn worden verder geanalyseerd. Bijvoorbeeld aandelenkoersen: alleen onder bepaalde condities wordt actie ondernomen. Werkt u met dat soort technieken?
Myczkowski: 'Ik zou er eens naar moeten kijken. De uitdaging die we hebben is in feite omgekeerd. Je kunt nooit vrijwel real time alle betekenis uit informatie halen. Tot voor kort vatten mensen in verband met de enorme storage eisen de data samen. Eerst werden de fysieke data opgeslagen bijvoorbeeld bij telco's en daaraan werden daaruit de data met betrekking tot de gespreksduur et cetera geëxtraheerd en opgeslagen.'

Maar dat gebeurt uiteindelijk ook, alle data wordt wel opgeslagen.
Myczkowski: 'Dat klinkt dan wel weer interessant. Binnen mijn vakgebied wordt *incremental learning* steeds belangrijker: hoe leer van je van een stroom data? Want datamining lijkt op het maken van een foto: het reflecteert de situatie op een zeker moment in de tijd. Maar een baby ziet er een week later al heel anders uit, terwijl het voor een volwassene een paar jaar kan duren voordat je uiterlijk verandert, de tijdschalen voor verschillende gebeurtenissen zijn anders. Het zou heel aardig zijn wanneer je een technologie zou kunnen vinden die de foto constant verandert, met nieuwe data. Niet alles hoeft te veranderen, de contour van je gezicht, de kleur van je haar meestal niet. Het leren van streams is de heilige graal in machinaal leren. Er zijn technologieën die dat mogelijk maken, het bouwen van modellen op basis van streams op een incrementele manier. Bij andere technologieën is het weer heel moeilijk. Het heeft wel een hoge relevantie: steeds meer systemen zijn stroom-gebaseerd, geen grote aantallen data die eenvoudig in de database gedumpt worden, maar stromen waarop je snel moet reageren. Het beste voorbeeld noemde je al, de aandelenmarkt. Daar kun je geen foto's maken maar moet je iets al doende leren. Er zit ook een principiële vraag achter: wat betekent incrementeel leren? Persoonlijk ben ik optimistisch: er zijn nogal wat trends die erop wijzen dat verbeteringen mogelijk zijn. Daar willen we naartoe: het leren van streams in real time en een systeem dat het mogelijk maakt dat te doen. Het is niet voor iedereen noodzakelijk, maar in alle gevallen waarin je *real time* interacties hebt zou het heel mooi zijn wanneer je dit tot je beschikking hebt. In ieder geval is dat de richting die we ingaan, we investeren daarin.'

Biologie

Aan het eind van ons gesprek – de opnameapparatuur was al ingepakt – begint Myczkowski te vertellen wat zijn achtergrond is. Hij blijkt bioloog, wat ook verklaart waarom de datamining-toepassingen op het gebied van de biowetenschappen hem fascineren.
Myczkowski: ´Ik ben geïnteresseerd in biologie omdat het een grote impact heeft op mensen en de problemen die we proberen op te lossen. Middelen tegen ziekten of interessante patronen hoe leven verandert is niet alleen interessant, maar het betreft ons allen. Voor mij als deel van de computerindustrie is dat het een wetenschap is die waarschijnlijk de grootste vraag naar IT zal hebben. Als je ernaar kijkt ziet het er heel ouderwets uit: een nat lab. Maar als je kijkt naar de potentie, het moet geautomatiseerd worden. Het grootste succes van de hedendaagse biologie is het ontraadselen van de volgorde van het menselijke genoom. De belangrijkste reden waarom dat zo snel kon, was dat we het proces geautomatiseerd hebben. Het proces was vanaf het begin gecomputeriseerd, er was een systeem dat 24 uur per dag draaide, 365 dagen per jaar, dat zich vrijwel uitsluitend met die ontcijfering bezig hield. Als je daarover nadenkt is dit wat automatisering voor hen bereikt heeft. Een vriend en collega van mij is tevens hoofd van de cancer genomics groep bij MIT. Zijn baan is het om datamining te gebruiken om te kijken welke genen een rol spelen en in welke types kanker. Gericht op die genen kun je dan weer geneeswijzen vinden. Dat is het gebruik van IT en gevorderde analytische technieken. Dit gebied zal steeds meer van die technologie gaan gebruiken en lettend op wat er nodig is, zal er een explosie in de grootte van data daar plaatsvinden. In de gezondheidszorg geldt hetzelfde, in de VS is het gezondheidssysteem een *basket case*. Het zou veel kunnen profiteren van BI, automation, zodat iedere keer dat je naar een arts gaat hij niet door duizend pagina's moet maar onmiddellijk kan vinden wat hij nodig heeft. Ik denk dat IT een enorme hulp zou kunnen zijn en dat we tot nu toe alleen maar nog niet eens echt begonnen zijn dat te exploreren. Voor een bedrijf als Oracle – en niet alleen Oracle – is er heel veel zinnigs te doen op dit gebied.'

Tekst en fotografie: **Dré de Man.**