

Een strenge database die toch nooit 'nee' zegt

# De Polis Papers (2): The Good, The Bad and The Ugly

René Veldwijk en Jeroen Schaay

**In het vorige artikel introduceerden we de Polisadministratie als een enorme database met een perfect geheugen. De PA is qua grootte bijzonder, maar niet zonder meer uniek. Wel uniek is de oplossing die is gekozen voor hét grote probleem van de loonaangifte, de administratie van gegevens waar een vlekje aan zit. De PA registreert niet alleen gegevens, maar controleert en registreert ook de kwaliteit van die gegevens en gebruikt deze informatie om gebruikers en afnemers op maat te bedienen.**

Elke serieuze database bevat foutieve gegevens. Deels gaat het om gegevens die niet *kunnen* kloppen, oftewel gegevenscorruptie. Gegevenscorruptie wordt voorkomen door het gebruik van database constraints en inputvalidaties. Maar dat is theorie: alle serieuze databases zijn min of meer corrupt en helemaal wanneer de gegevens van buiten de organisatie komen. Bij de PA was (en is) dit probleem door een ongelukkige samenloop van omstandigheden extra groot en het symptoom was gegevensuitval. Medio 2007 bleek ruim 70 procent van de loonaangiften over 2006 onbruikbaar voor de Belastingdienst, grotendeels door uitval in de lange gegevenslogistieke keten tussen werkgevers, Belastingdienst en UWV. Wat een lekvrije gegevenspijp had moeten zijn, leek eerder op een vergiet.

Om te begrijpen hoe deze situatie kon ontstaan, moeten we kijken naar de opzet van de loonaangifteketen. Werkgevers doen aangifte via honderden salarispakketten; antiek en modern, groot en klein, pakket en maatwerk, in huis en in service. Iedereen moest vanaf 2006 complexe XML loonaangifteberichten van soms enorme omvang opsturen naar de Belastingdienst, die deze gegevens met diverse systemen verwerkt. Voor de Belastingdienst staat bij de ontvangst de incasso centraal en daarvoor heb je alleen de totaalbedragen uit het XML bericht nodig. Deze collectieve gegevens worden uit het XML bericht 'gestript'. Is alles goed, dan wordt de rest, inkomensgegevens van individuele personen, doorgezonden naar het UWV. Zoniet dan houdt de Belastingdienst het bericht vast tot het probleem is opgelost. De gegevens die UWV binnenkrijgt zijn niet altijd in volgorde van verzending door de werkgever of verwerking door de Belastingdienst. UWV stopte deze berichten in een *staging*

database, daarna in een 'voordeurapplicatie' en uiteindelijk in een Polisadministratie systeem. En elke transformatie leidde tot uitval.

Dat deze keten niet werkte was in essentie op drie oorzaken terug te voeren: de lengte van de keten, het ontbreken van snelle en goede terugkoppeling van gegevensfouten en de verschillende eisen die de gebruikers van de Polisadministratie aan de gegevens stellen. De keten werd ingekort doordat UWV de drie systemen binnen haar domein verving door één integraal nieuw PA systeem. De terugkoppeling van fouten werd verbeterd doordat de Belastingdienst voor zover mogelijk bij ontvangst ook ging controleren op gegevens die niet direct van belang zijn voor de belastingheffing. Maar elke aangifte die voldoet aan een aantal basale controles komt in de keten. Fouten worden daarna pas minimaal een maand later of (nog) helemaal niet teruggemeld aan de werkgever. En dus moest de PA database bestand zijn tegen *alles* dat fout kan gaan. Het gaat bij de PA niet alleen om *Total Recall* maar ook om *Total Permissiveness*. De nieuwe PA moest zelfs bestand zijn tegen vergaand versjeteerde XML berichten. Het motto: "Goed dat de Belastingdienst berichten controleert aan de poort, maar alles wat daar voorbij komt mag verderop in de keten niet meer uitvallen".

## Ugly data: de NecroPolis

Het idee van een *Permissive Database* is een contradictie. Databases bevatten gestructureerde data ofwel data die zich aan regels houden. Een loonbedrag komt in een database als *number(9,2)* en het DBMS garandeert daarmee dat er alleen gegevens worden geregistreerd die numeriek zijn. Je kunt natuurlijk bedragen opslaan in, zeg, een *Varchar(4000)* veld, maar wie dat doet mag heel veel programmeren en krijgt een systeem dat beroerd performt. De oplossing die we voor de PA hebben gekozen is onderscheid maken tussen *good*, *bad* en *ugly* data. Die laatste categorie data komt niet in de PA maar valt evenmin uit. De PA bezit een deelschema, NecroPolis genaamd, waarin alle berichten worden opgeslagen die niet of niet helemaal kunnen worden verwerkt. De structuur van deze uitval-database volgt de XML structuur van het loonaangiftebericht, zie afbeelding 1.

Bijna alle berichten zijn te verwerken – wat niet wil zeggen dat de data correct zijn. Is verwerking mogelijk dan blijft de NecroPolis leeg. De berichtgegevens worden dan geconsolideerd

naar standen-in-de-tijd en opgeslagen in de database die we in het voorgaande artikel hebben beschreven. Is een bericht niet voor 100 procent verwerkbaar dan verdwijnt het in de NecroPolis – inderdaad een database met alleen maar *Varchar*-velden voor de opslag van berichten. De NecroPolis bevat daarnaast berichten die technisch wél correct zijn, maar irrelevant. Het gaat dan om gevallen waarin een werkgever eerst bericht A instuurt en daarna een verbeterd bericht B. Als die berichten binnenkomen in de volgorde van verzending, dan worden alle gegevens opgeslagen en zijn beide situaties terug te vinden door te reizen in de transactietijd (zie afbeelding 3 in het vorige artikel). Als de Belastingdienst echter de volgorde verwisselt dan verdwijnt bericht A als mosterd na de maaltijd in de NecroPolis. Op deze wijze bereiken we met de PA een nog hoger niveau van *Total Recall*: we weten niet alleen alles wat er gebeurd is met de gegevens die in de kern-PA zijn opgenomen, maar kunnen ook terug naar alles wat er aan XML berichten is aangeleverd. De NecroPolis sluit de strenge PA database aan op de *alles-kan-in-principe* wereld van de loonaangifteketen. Pas met de NecroPolis wordt de PA een echte *Total Recall* database. Merk op dat de NecroPolis niet los staat van de kern-PA van het vorige artikel. We kunnen vanuit de NecroPolis de link leggen naar de kern-objecten van de PA – Werkgevers, Personen en Inkomstenverhoudingen – mits de identificerende gegevens herkenbaar zijn. In afbeelding 1 is dat aangegeven met de tekst "FK": een *foreign key* die we niet op de database kunnen leggen omdat een klein deel van de gegevens corrupt is.

## Intermezzo: corruptie is een keuze!

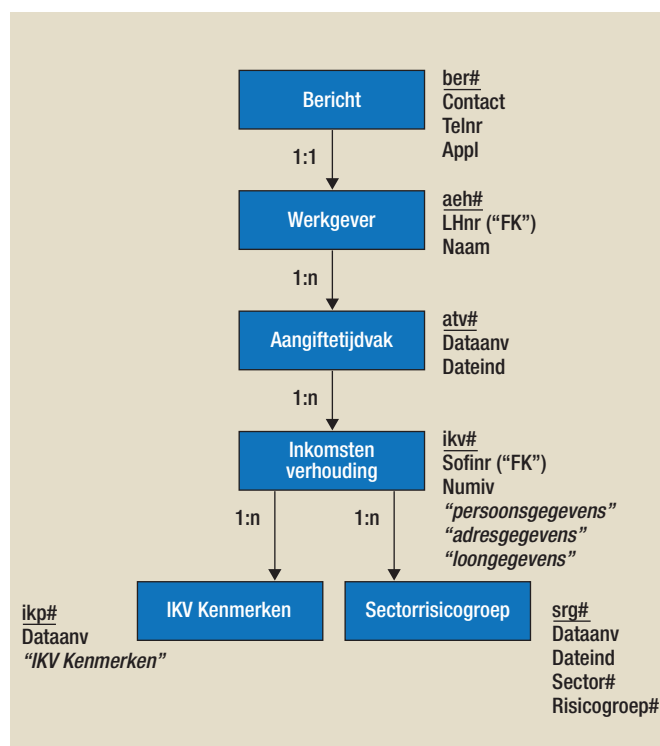
Zoals gezegd komt maar een fractie van de binnenkomende berichten in de NecroPolis terecht. Maar met de gegevens die de PA wel halen is ook vaak iets mis. De meeste van deze 'data-met-een-vlekje' vielen in de oude loonaangifteketen vaak uit. Als uit het oude systeem gegevens werden geleverd dan waren dat er veel te weinig, maar gegevens die werden geleverd waren perfect. Bij onze PA is dat omgekeerd: we hebben alles in huis, maar we de afnemer die argeloos gegevens uit de PA over haalt. We hebben het probleem van gegevens met een vlekje namelijk niet opgelost maar verplaatst naar de PA. Leveren we die data uit dan kan de afnemer onaangenaam worden verrast. We moeten voor een echte oplossing daarom nog een stap verder weg van het klassieke denken over databases. In dat denken bevat een database misschien wel foutieve maar geen inconsistente (corrupte) gegevens. Onze PA is wel corrupt en dat is een bewuste keuze. Het logische vervolg is dan dat we afstappen van de gedachte dat *wij* het zijn die bepalen wat goed is en wat niet. We laten dit over aan de afnemers en beperken ons tot het signaleren van mogelijke vlekjes op de gegevens – wellicht op verzoek van die afnemer zelf. En het is aan onze afnemers om te zeggen of ze een signalering belangrijk vinden. Is men zwaar op de hand dan leveren we de bevlekte data niet uit. Wel melden we het aantal objecten dat we niet hebben uitgeleverd. Is een signalering oninteressant of mag de afnemer deze niet zien, dan

leveren we de gegevens gewoon uit. Het interessantst is echter de situatie waarbij de afnemer zowel de gegevens wil hebben als de signaleringsinformatie. Die informatie kan worden geleverd in XML berichten of in een bulkbestand dat we FTP'en. Maar het kan ook dynamisch worden getoond op een raadpleegscherm. Afbeelding 2 toont een geanonimiseerd maar verder live, extreem geval uit de PA database.

## Bad data: Constraints en Signaleringen

Alle gegevens waarop één of meer signaleringen betrekking hebben en die de afnemer wenst te zien, krijgen daadwerkelijk een vlekje. Komt de cursor op zo'n bevlekt veld dan wordt de gegevensanomalie getoond in een *balloon*. Hier hebben we alle *balloons* tegelijk afgebeeld – excuus voor het resulterende 'discoscherm'. Kijken we naar de tijdgegevens dan zien we dat beide tijdlijnen corrupt zijn. Dit is een ernstige fout, niet gemaakt door de werkgever of de Belastingdienst, maar door onze laad-programmatuur. Op basis van onze High-T datadictionary is *on the fly* bij het tonen van de gegevens vastgesteld wat er mis is. Omdat we zijn aangelogd als gegevensbeheerder krijgen we deze gegevens te zien. Performt dat op een Terabyte database? Jazeker, al moet je het wellicht niet met duizenden gebruikers tegelijk doen. Het is iets voor *content* beheerders.

Tot nu toe zien we standaard functionaliteit van onze High-T omgeving waarmee de PA is ontwikkeld. Kijken we vervolgens naar de stapel meldingen die betrekking hebben op de rubriek RdnExFlex (Reden Einde Flexverhouding) dan komen we bij de toevoegingen voor de PA terecht. Bij het laden van de gegevens is vastgesteld dat er hier een aantal signaleringen moet worden



Afbeelding 1: De NecroPolis.

**PERSONENDASH\_BSN - Personendashboard (BSN-ingave)**

Ingave burgerservicenummer (BSN\_INGAVE\_DS) (1)

Burgerservicenummer: 940000003 | K Jong Ding

**Overzicht Inkomstenverhoudingen (LA\_IKVD1) (1)**

Sofinr	Naam	IKV Id	LHnr	Naam Adm.Eenh.	NumIV	Pers. nr	Sexe	DatAanvTvk	D.
1	940000003	Jong Ding, K	303733	123456789L01	Reservoir Dogs Restaurant	1	6027635	2	

**Violations:**  
- R:Lopend fraudeonderzoek.

Inkomstenperiode: Historie (0) | AVB (0) | Inkomsten-en-perioden | Signaleringen | Logging | Transacties | Laad issue  
 Inkomstenopgave | Sector-Risicogroep

Naamhistorie (2) | Adreshistorie | Geldigheidshistorie | Sleutelhistorie

**Inkomstenverhouding geldigheid historie (LA\_IKVHISTD4) (6)**

	DatAanv	Dat tot	RdnExFlex	Ttime in	Ttime eind
1	20060814	20070423	3	11-11-2007 11:11:11,666666	07-03-2008 04:01:04,266000
2	20060814	20070101	3	07-03-2008 04:01:04,266000	31-12-9999 00:00:00,000000
3	20070101	20070129		07-03-2008 04:01:04,266000	31-12-9999 00:00:00,000000
4	20070129	20070423	3	0	
5	20070521	20070618	1	1	
6	20070618	99991231	3	11-11-2007 11:11:11,666666	31-12-9999 00:00:00,000000

**Violations:**  
- C:Constraint HIS\_IKV\_IKGH03 violated: Er mag geen overlap tussen de transactietijden in de historie zijn.

**Violations:**  
- R:Code reden einde flexwerker ongeldig op datum aanvang.  
- R:Code voor reden einde IKV flexwerker onbekend.  
- R:Code einde inkomstenverhouding flexwerker alleen gevuld voor flexwerker en einddatum gevuld (ex 0 en 00)  
- R:Code einde inkomstenverhouding flexwerker alleen gevuld voor flexwerker en einddatum gevuld.  
- C:Domain constraint RDNEINDFLEX violated, invalid value 3

**Violations:**  
- C:Constraint HIS\_IKV\_IKGH02 violated: Er mag geen overlap tussen de bestaanstijden in de historie zijn.

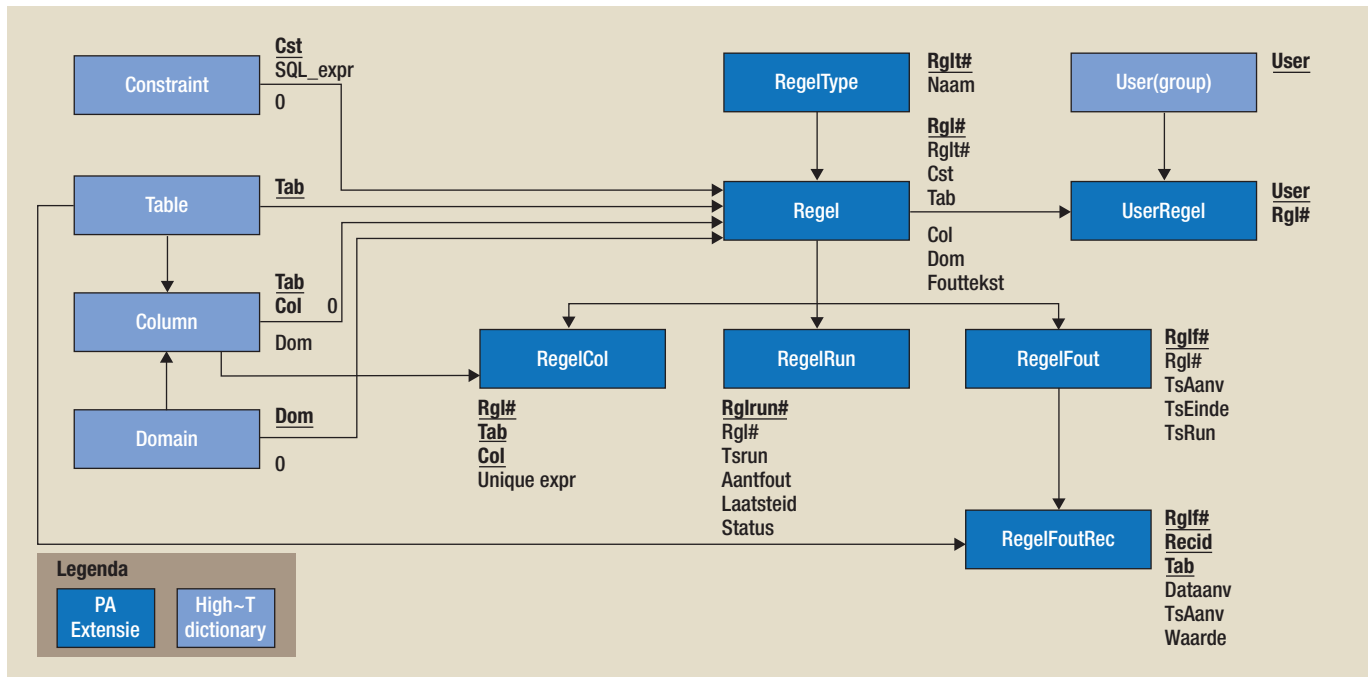
**Afbeelding 2:** Data met een vlekje (gegevensanomalieën).

geplaatst en deze signaleringen zijn fysiek opgenomen in een signaleringsdatabase. Net als de Necropolis maakt deze onderdeel uit van de PA. Voor de gebruiker is er geen verschil – die ziet vlekjes en balloons. In die balloons wordt wel vastgelegd of er sprake is van een *on the fly* geconstateerde database constraint (C – constraint) of van een signalering (R – rule). UWV-medewerkers of afnemers krijgen in beginsel alleen de laatste te zien en dan nog alleen wanneer die voor hen relevant zijn. Het definiëren van een constraint is een fluitje van een cent: schrijf een SQL statement, voeg een aantal markers en stuurdata toe en alles werkt. Het plaatsen van signaleringen is daarentegen een flinke klus. Signaleringen moeten daadwerkelijk worden opgespoord en opgeslagen in een database en daarna moet de laadprogrammatuur worden uitgebreid om signaleringen bij te plaatsen. (Merk op dat in een *Total Recall* database signaleringen nooit worden verwijderd.) We doen dit alles maar om één reden: performance. We kunnen eenvoudigweg niet bij

een multi-gigabyte levering *en passant* allerlei controle-query's op de enorme PA database laten lopen. Hebben we eenmaal de mogelijkheid om signaleringen te plaatsen dan wordt het feest. Zo zien we dat er een fraudeonderzoek loopt naar de opgevraagde arbeidsverhouding. Hier zijn we het signaleren van gegevenscorruptie voorbij en betreden we de wereld van de onjuiste data, de wereld van de gebruikers. Deze signalering komt dus niet uit de laadprocessen maar is handmatig geplaatst door de fraudebestrijders van UWV. Uiteraard wordt deze signalering niet zomaar aan alle afnemers van de PA getoond. Eerlijk gezegd hebben we deze signalering zelf geplaatst. De PA is nog geen jaar operationeel en deze vormen van geavanceerd gebruik staan nog in de kinderschoenen.

### Bad data voor technuten

Het signaleringsmechanisme is specifiek ontwikkeld voor de PA, maar is toepasbaar voor elke database. Het enige dat het PA



**Afbeelding 3:** PA signaleringen als generieke extensie op de High~T dictionary.

signaleringsmechanisme toevoegt is een manier om signaleringen helemaal generiek op te slaan. Afbeelding 3 toont het gegevensmodel, deels van de High~T dictionary en deels van de uitbreiding voor de PA.

Basale dictionary tabellen als Table, Column en Domain vormen de basis voor de registratie van signaleringen. RegelType bevat een klasse van regels die we in één generieke expressie kunnen vangen. Regel bevat de individuele soorten signaleringen van een RegelType die worden geparаметriseerd door middel van verwijzingen naar objecten in de High~T dictionary. In RegelCol wordt vastgelegd welke kolommen uit het datamodel een rol spelen bij de signalering. Per gevonden signalering wordt één record aangemaakt in RegelFout. Indien kolommen uit verschillende tabellen een rol spelen, wordt in RegelFoutRec een record aangemaakt voor iedere unieke tabel. De verwijzing naar het precieze instance record in een PA tabel gebeurt door RegelFoutRec.RecId, eventueel aangevuld met *DatAanv* en/of *TsAanv* indien het een historische tabel betreft. RecId is een numeriek veld en onvermijdelijk eist het signaleringsmechanisme dat de tabel waarop signaleringen worden geplaatst een numeriek veld heeft dat als *unique key* functioneert. Voor 'schone' records mag dat veld leeg zijn, dus eventuele aanpassingen aan de database blijven minimaal. UniqueExpr bevat de naam van de identificerende kolom waarnaar verwezen wordt door RegelFoutRec.RecId of levert deze kolomnaam op.

Tenslotte wordt in RegelRun de voortgang van een signaleringsrun bijgehouden zodat deze herstartbaar is. Er zijn namelijk regels die los van de laadprocessen lopen en dagenlang kunnen draaien.

In de polisdatabase worden alleen gegevens bijgeladen en al deze gegevens hebben een transactietijdlijn (zie vorig artikel).

De laatste succesvolle TsRun van een regel in combinatie met RegelCol geeft dus precies aan welke gegevens wel en welke (nog) niet gecontroleerd zijn. Omdat ook het signaleringsmechanisme *Total Recall* is kunnen we signalen ook achteraf doorgeven aan de afnemers van PA gegevens. Tenslotte kunnen regels met behulp van UserRegel toegewezen worden aan (groepen) gebruikers. Redelijk uniek, zo menen wij.

## We have a dream!

Actief gebruik van signaleringen door de gebruikers van de PA zoals in afbeelding 2 is op dit moment nog beperkt. Reden daarvoor is vooral dat de webservices waarmee de PA wordt benaderd al waren ontwikkeld voor de wereld van de perfecte data. Omdat de specificaties niet voorzien in het meeleveren van signaleringsinformatie, wordt de keuze voor onze afnemers nu vaak beperkt tot ofwel corrupte gegevens zonder bijsluiter opvragen ofwel een *no data found* melding terugkrijgen. Beide situaties komen voor. De tienduizenden werkers bij UWV, CWI (nu UWV Werkbedrijf) en Gemeenten krijgen alle PA gegevens, *good and bad*, ongefilterd en ongemarkeerd door. En zelfs dat heeft heel wat voeten in de aarde gehad: op drie plaatsen moesten XSD's worden uitgekleeft, waarna er nog geen gegevens doorkwamen omdat alle controles ook nog eens hard waren uitgekraast in Java. Zelf kunt u naar verwachting nog dit jaar de omgekeerde situatie meemaken wanneer u in de gelegenheid wordt gesteld om met uw DigiD uw loongegevens te bekijken in de vorm van een digitaal verzekeringsbericht (DVB). U zult dan evenmin data met vlekjes zien, want elk record waarmee iets mis is houden we noodgedwongen achter. Als u straks inkomensgegevens mist dan wil dat nog niet zeggen dat we die gegevens niet hebben. We hebben ze bijna altijd wel in huis en we weten

---

wat er mis mee is. We kunnen het alleen niet tonen. We mogen hopen en verwachten dat het allemaal beter gaat worden, al zal het even duren voordat elke afnemer zich het 'vlekkenbeginsel' heeft eigen gemaakt. Als straks duidelijk wordt wat het signaleringsmechanisme allemaal kan, dan zullen afnemers, werkgevers en burgers willen zien wat er volgens UWV en anderen mis is met hun gegevens. Terugkoppeling in maanden wordt dan terugkoppeling in dagen en de kwaliteit van de PA zal met sprongen omhoog gaan. Kijk nog één keer naar afbeelding 2. Als de fraudebestrijders van UWV bepaalde signaleringen mogen plaatsen bij bepaalde gegevens, dan is het op termijn best denkbaar dat uw salarisadministrateur, uw pensioenfonds of uzelf dat ook doet; uiteraard binnen strikte, door de autorisaties van de PA afgebakende grenzen, maar verder zonder

administratieve rompslomp en natuurlijk 7x24. Dat is pas lastenverlichting. Een slimme toepassing van een mechanisme om gegevenscorruptie te managen leidt dan tot een systeem met de allerbeste gegevenskwaliteit. Komt tijd, komt kwaliteit!

*Het laatste woord is in deze serie nog niet gezegd over de mogelijkheden die het signaleringsmechanisme biedt. Maar eerst gaan we in de komende twee artikelen in op de manier waarop de PA uw persoonsgegevens beschermt en het gebruik ervan bewaakt. Ook daarbij gaan we met de PA nogal wat verder dan gebruikelijk.*

**René Veldwijk** en **Jeroen Schaay** zijn partner bij FAA Partners, onderdeel van de Ockham Groep.

---

## Vervolg van Business Objects XI R3.1

---

hoeft dat ook niet omdat de bestaande producten nog tot en met 2016 ondersteund zullen worden, maar er gaat wel het een en ander veranderen. Werd er enkele maanden geleden nog uitermate vaag gedaan over de toekomst van Bex en BW, inmiddels is duidelijk wat er gaat gebeuren. De volledige BI Roadmap-presentatie kunt u downloaden van de DB/M site, maar hier volgen alvast de highlights:

- SAP's WebApp designer verdwijnt en Xcelsius wordt de tool voor dashboards en visualisatie;
- SAP's Report designer verdwijnt eveneens en Crystal Reports wordt de enterprise reporting tool;
- SAP had geen web-based ad hoc query-tool, dus WebIntelligence gaat dit gat opvullen;
- Dan het grootste nieuws: de Bex Analyzer verdwijnt als stand-alone product en wordt samengevoegd met Voyager in een nieuw analyse-tool genaamd 'Pioneer';
- Het Enterprise Data Warehouse platform is en blijft SAP BW, aangevuld met de BW Accelerator;
- Als BI-platform blijven zowel Netweaver BI als BO Enterprise naast elkaar bestaan. De eerste uiteraard voor (bestaande) SAP-omgevingen, de tweede als applicatie-agnostisch BI-platform. Er komt wel steeds meer integratie tussen de twee. Zo zal het in 2010 mogelijk moeten zijn om met Netweavers Visual Composer Xcelsius componenten onder te brengen in Netweaver-oplossingen;
- Last but not least wordt er gewerkt aan project 'Newton'. Een uitbreiding van het platform dat ook als stand-alone analyseproduct gebruikt kan worden. Lees 'in memory data marts', 'high performance', 'write back capabilities', 'ranking', 'top/bottom N analysis'. Er is geloof ik ook een Zweeds/Amerikaans bedrijf met een Q dat iets dergelijks levert en daar zeer succesvol mee is, maar zo moest ik dat volgens Timo Elliot niet zien. Right!

### Conclusie

In een paar pagina's is het vrijwel onmogelijk om het complete aanbod van een leverancier als SAP-BO de revue te laten passeren. De ontwikkelingen op het gebied van bijvoorbeeld Data Services en Master Data Management houdt u dus nog tegoed, evenals die van text-analyse, visualisatie, data mining en het on-demand aanbod. Ook de 'Edge' bundel voor de mid market en alle producten voor financial- en corporate performance management hebben we hier niet kunnen bespreken. Het is wel duidelijk dat men nog veel werk te verzetten heeft, met name om alle producten goed te integreren. In elk geval is XI 3.1 op het gebied van platform-, versie- en gebruikersbeheer een flinke stap in de goede richting. Hetzelfde kan gezegd worden over de laatste roadmap waarin duidelijk keuzes voor de toekomst zijn gemaakt, waardoor veel onzekerheid bij bestaande en potentiële gebruikers wordt weggenomen.

De vraag is of dat helemaal gaat lukken: tijdens de laatste Gartner BI Summit in januari van dit jaar scoorde BO van de vier MISO leveranciers de meeste 'Caution' en 'Promising' scores en stak ten opzichte van met name IBM wat bleekjes af. De Magic Quadrant van 16 januari jongstleden liet ook al een verslechtering zien ten opzichte van de grootste concurrenten. Maar goed, zelfs als men alleen succesvol is (blijft) op de SAP thuismarkt ziet de toekomst er rooskleurig uit, en met 26 procent marktaandeel volgens dezelfde Gartner Group blijft BO nog steeds de te kloppen partij.

### Jos van Dongen

Jos van Dongen (jos@tholis.com) is onafhankelijk adviseur, auteur en spreker. Met speciale dank aan: Niko Brouwer, directeur van VCD Business Intelligence, Timo Elliot, Business Objects VP Strategic Marketing, Coen de Koning, Business Objects Pre-sales Consultant en oppergoeroe.