

VELDWIJK

# Doodsbericht

Scène I. Een paar jaar geleden was ik betrokken bij een post mortem op een ICT-project. Het ging om een systeem voor het opsporen van witte fraudeurs, mensen met een uitkering die ook een wit baantje hebben. De kernfunctie van het systeem bestond daarin dat bestanden uit diverse bronnen met elkaar werden vergeleken om bij records met eenzelfde sofinummer een zogenaamd samenloopsignaal af te geven. In de ICT-aanpak van de jaren zeventig zou dit zijn opgezet als een *merge* op sequentiële bestanden. In de jaren tachtig zou een *inner join* op tabellen met dezelfde primaire sleutel zijn uitgevoerd. In de jaren negentig zou sprake zijn geweest van een eenvoudig *datawarehouse*. Maar dit systeem was gebouwd op basis van een *message handling* paradigma. Het klassieke bestand is vervangen door een verzameling losse gebeurtenissen die elk voor zich een match-proces *triggeren*. Dat men op maandbasis bestanden binnen kreeg was geen reden om gegevens niet per seconde te verwerken. Dat message handling intrinsiek veel complexer is dan *batch handling* boeide evenmin. Helaas bleek dit eenvoudige systeempje te moeilijk om te bouwen.

Scène II. Samen met een collega doe ik een audit op een innovatief salarissysteem dat onvoldoende performance biedt. Het innovatieve karakter heeft betrekking op *near real-time* verwerking van transacties. De gedachte is dat een mutatie van bijvoorbeeld een bruto salaris direct wordt doorgerekend naar netto bedragen, die dan meteen kunnen worden teruggestuurd naar de medewerker achter het scherm. Naast een aantal andere knelpunten vinden we als kernprobleem dat men niet volledig afscheid heeft genomen van het batch-karakter van veel processen. Onze conclusie is dat oplossing van de problemen moet worden gezocht in een radicale keuze voor berichtverwerking en in het elimineren van elke batch waarin wordt gerekend. Het probleem komt namelijk door een keuze voor de slechtste van twee werelden: de datastuwmeren van batches en de traagheid van berichtverwerking. Deze voorbeelden laten zien dat er een voor administratieve omgevingen nieuw paradigma rondwaart. Bestanden zijn uit, berichten in. Scène I laat zien dat het nieuwe paradigma er soms met de haren wordt bijgesleept (*what's new?*). Scène II laat zien dat je niet ongestraft paradigma's kunt mixen. Ook dit is *déjà vu*: zie de ellende met workflow in administratieve systemen. Bij het salarissysteem was er wel degelijk een argument voor berichtverwerking en die gevallen zijn er te over. Denk bijvoorbeeld aan directe verwerking van banktransacties. Merk daarbij ook op dat de toename van directe berichtverwerking alles te maken heeft met (mobiel) Internet. Echter, er blijven veel gevallen waar berichtverwerking geen enkel probleem oplost maar wel creëert.

Over naar scène III. Ik ben betrokken bij de totstandkoming van een enorme bak – Terabyte plus – met gegevens over inkomens. Op die bak lopen grootschalige mutatie- en afnameprocessen. Het berichtenparadigma is in deze omgeving volledig dominant. Alle inkomende en uitgaande stromen moeten in hollebolle XML-vorm plaatsvinden. Batch-verwerking is onmogelijk gemaakt door de informatie over wie wat geleverd moet krijgen in andere systemen op te slaan. Zodoende zijn we veroordeeld tot het afwachten van atomaire XML-*requests* waarop we atomaire XML-*responses* geven. Sommige van onze afnemers krijgen maandelijks vele Gigabytes aan data uit het systeem en dat is dan exclusief de XML-overhead. Werken met FTP is verboden want strijdig met het berichtenparadigma.

Je zou zeggen dat een project als dit gedoemd is te mislukken, maar dat is niet zo. De eisen die gesteld worden zijn namelijk totaal onmogelijk, terwijl berichtverwerking, net als bij het systeem voor witte fraudeurs het berichtenparadigma, geen nut heeft. De omvang van het systeem en de gegevensstromen zijn gewoon te extreem. Een batch die een plat bestand uitspuwt is zeker 100 keer sneller dan een XML-dialogue en dat maakt het verschil tussen werkbaar en niet werkbaar. Het mooie van bestanden boven berichten is ook dat met de bestanden pakbonnen kunnen worden meegestuurd met voor de afnemers nuttige controle informatie. Misschien nog wel het aardigst van alles is dat de realisatiekosten van zo'n klassiek systeem maar een fractie zijn van de kosten van een moderne berichtengehaktmolen. Om te beginnen doet bij een goed opgezet batch-georiënteerd systeem de database server het grote werk, waar bij een berichtgeoriënteerd systeem de verwerking verspreid is over allerlei software- en hardware-componenten. Bij grote systemen loopt dat verschil in de papieren. Veel belangrijker is dat ontwerp en ontwikkeling door verschillende redenen vele malen – in mijn ervaring circa 10 keer – goedkoper is dan volledig berichtgeoriënteerd werken. Eind goed al goed: iedereen erkent dat klassieke batch-verwerking hier de aangewezen weg is en het systeem komt van de grond. Conclusie: berichtgeoriënteerd werken kan nuttig zijn en is steeds vaker noodzakelijk. Wie er aan begint doet er wel verstandig aan om eerst goed stil te staan bij de enorme voordelen van klassieke vormen van automatisering. Ik heb mijn laatste post mortem nog niet uitgevoerd en mijn laatste column hierover niet geschreven.

**René Veldwijk**

Dr. R.J. Veldwijk ([rene.veldwijk@faapartners.com](mailto:rene.veldwijk@faapartners.com)) is partner bij FAA Partners, een onderdeel van de Ockham Groep.