

Ik ben van de generatie die vooral speelde met fysiek speelgoed: treintjes. Meisjes speelden met poppen, jongens hadden een treinbaan. Het grove werk: schaal HO, of het fijne spul: schaal N. Inmiddels hebben we de virtuele hedendaagse evenknie van de treinbaan. Mede mogelijk gemaakt door SOA en mashups.



Treintjes voor volwassenen

In de Oracle-wereld valt het woord Java inmiddels al sinds medio 1998, toen het Java-ontwikkelgereedschap van Borland – JBuilder - in licentie werd genomen. Een jaar of drie later herschreef Oracle JBuilder en het resulterende tool, JDeveloper, was voortaan 'the way to go'. Maar overstappen op een andere programmeertaal was niet waar het om ging. Zeker in de begindagen van overstappen op Java bleek al snel dat ontwikkelen met Java neerkwam op 'ga terug naar start'. Met Oracle Forms en Oracle Designer werden heel veel programmeeruitdagingen impliciet getackeld. Met Java moest dit alles weer opnieuw uitgevonden worden en van scratch af aan geprogrammeerd.

Bij veel traditionele Oracle-ontwikkelaars kreeg (en soms nog altijd: krijg) je amper uitgelegd waarom Java toch echt z'n pluspunten heeft. En dat is niet zo gek. Het is niet zozeer de taal die het verschil maakt. Het is het paradigma. Vlak nadat Java z'n intrede deed maakte ook de term Service Oriented Architecture z'n opwachting. En dat maakte wel het verschil. Strikt genomen kun je met Java prima een monoliet maken zoals dat voorheen met Oracle Forms ook prima ging. Maar als je Java programmeert met de concepten van een SOA, dan wordt het een ander verhaal. Herbruikbare bouwsteentjes van functionaliteit. Die kunnen worden gebruikt door verschillende applicaties. Daar leende Oracle Forms zich toch minder voor.

Inmiddels zijn we weer tien jaar verder en het SOA paradigma is wijd verbreid. Maar nog altijd redeneren veel bedrijven vanuit het concept van de vroegere monolieten. Als een van hun belangrijke applicaties op de rol staan om met pensioen te mogen, dan wordt er vaak een applicatie gekozen die exact dezelfde functionaliteit biedt als de vroegere legacy-applicatie. Geloof me, dan hoeft de applicatie in kwestie echt niet in Java te zijn geprogrammeerd. Maar dat is dan wel een gemiste kans. Als je anno nu een applicatie selecteert of ontwikkelt, dan is de truc om te kijken of de functionaliteit die je nodig hebt niet al geheel of gedeeltelijk als service wordt geleverd buiten de grenzen van je eigen bedrijf. En of de functionaliteit die de applicatie zelf biedt ook als services kunnen worden gepubliceerd. Want hoe meer services je toevoegt aan je SOA, des te

lager worden de investeringen in de toekomst. Je hoeft immers geen monoliet meer aan te schaffen, alleen de resterende services.

Uiteraard is er een aanleiding voor deze SOA-hersenspinsels. Onlangs viel mijn oog op een prachtige toepassing van SOA, mashups en open standaarden. Een voorbeeld dat ongelooflijk goed illustreert wat ik hiervoor heb beschreven. Waar we staan en wat anno 2011 mogelijk is. Google even op 'mailspool' en selecteer de hoogst gerankte hit die wordt weergegeven: Actuele Spoorkaart Nederland. Wat je hier ziet is een prachtig stukje programmeervlijt van een hobbyist. Eerst nog even het schrille contrast: de site van de NS biedt ook een actuele spoorkaart. Dit initiatief komt echter niet verder dan het kunnen aanklikken van een station waarna de treintijden worden getoond op dat station.

Nee, dan de Actuele Spoorkaart Nederland van de anonieme creatieveling. Het ziet er uit als een live opname vanuit een satelliet, waarin je alle wegen en spoorlijnen ziet van Nederland, in te zoomen tot ongeveer schaal 1:1000. En live betekent echt live: elke rijdende trein in Nederland zie je echt rijden. Het is een mashup van de open source kartografische data van OpenStreetMap (openstreetmap.nl) en de actuele treintijden (incl. vertraging) opgevraagd met een service bij de NS. Deze informatie is vervolgens grafisch geplot in de vorm van bewegende treinen op de kaart.

Ondanks de verbluffende look en feel en functionaliteit, denk ik dat de applicatie niet veel regels code hoeft te beslaan. Gewoon slim combineren van elders aanwezige services en data. Moderne user interface. En toch van beperkte omvang door maximaal hergebruik. Dát is nu softwareontwikkeling anno 2011.

Harold Gerritsen is directeur van A New Link bv, een bureau dat bedrijven adviseert, traint en coacht op het snijvlak van Oracle en Business. Daarnaast is hij ook zelf deeltijds actief 'in het veld', in de rol van Enterprise Architect. Hij is te bereiken via harold.gerritsen@anewlink.nl.