

De meeste architecten kennen de volgende situatieschets wel. Je wordt ergens naar binnen gehaald met de vraag om 'even' een nieuwe architectuur neer te zetten. De beoogde lange termijnvoordelen zijn het reduceren van kosten en het verhogen van de flexibiliteit voor de klanten. De organisatie heeft behoefte aan een modernere vorm van haar dienstverlening en tegelijkertijd een kortere time to market.

Recycle uw legacy met een SOA

De IT weer inzetten als strategisch voordeel

De nieuwe architectuur moet minder gericht zijn op systemen. De nadruk moet juist komen te liggen op de bedrijfsprocessen. Het is de bedoeling het adaptief vermogen van de business te verhogen. Het management heeft aangegeven dat het niet de bedoeling is om de gedane investeringen in de bestaande legacy systemen verloren te laten gaan.

Wat je in veel organisaties aantreft is een veelheid aan informatiesystemen die grotendeels bestaat uit maatwerksystemen, aangevuld met enkele standaardpakketten. Je nieuwe opdrachtgever is hierin geen uitzondering. Veel van de bedrijfslogica is meervoudig geïmplementeerd. Omdat de verschillende applicaties toegang moeten hebben tot de data van de business vinden replicatieslagen plaats. Daar waar replicatie niet mogelijk is, wordt de data in eigen applicatiedatabases gedupliceerd. Applicaties communiceren onderling via een wirwar aan point to point connecties.

De hierboven beschreven complexiteit heeft als gevolg dat het wijzigen van bedrijfsprocessen vaak erg tijdrovend is. De organisatie krijgt het gevoel dat zij langzamerhand de grip aan het verliezen is op de implementatie van de business. Een nieuwe architectuur moet helpen om de ontstane kloof tussen business en IT te vernauwen. De IT moet weer worden ingezet als strategisch voordeel.

Het bedrijf krijgt steeds meer te maken met externe dienstverlening in de vorm van webservices. Het is een kostbare aangelegenheid om deze webservices vanuit de legacy te laten aanroepen. Hiervoor zijn maatwerkadapters geschreven in de legacy. Het aan-

passen van een interface is altijd een tijdrovende en specialistische klus. Daarnaast vragen klanten ook steeds vaker om webservices, zodat ze de aangeboden dienstverlening kunnen integreren in hun eigen bedrijfsprocessen. Er is dus behoefte aan een architectuurstijl die dit op overzichtelijke wijze mogelijk maakt. Je denkt aan een Service Oriented Architecture, maar eerst even doorvragen hoe het staat met aan SOA-gerelateerde kennis binnen het bedrijf.

De afdeling IT heeft specialistische kennis van de programmeertalen waarmee de huidige backoffice legacy is gebouwd. Wat ontwerp betreft wordt de gestructureerde methode gehanteerd. Van objectoriëntatie is weinig tot niets bekend. Binnen het bestaande applicatielandschap bestaan met name client-server applicaties en batchgeoriënteerde programmatuur. Er lopen wel wat mensen op de afdeling met kennis van XML, maar er is geen kennis op het gebied van de programmeertalen die door de meest gebruikte SOA-omgevingen worden gebruikt. Daarnaast wordt duidelijk aangegeven dat het niet de bedoeling is om de reeds gedane investeringen in de bestaande applicaties overboord te gooien. Tevens mogen de lopende projecten niet worden gehinderd door het implementeren van de nieuwe architectuur.

Zo, een hele kluit dus, maar omdat je de uitdaging niet schuwt neem je de opdracht met een glimlach aan.

SOA

Hoewel de meesten wel weten waar een SOA voor staat, wil ik toch graag even in het kort mijn visie



Armand Wijskamp is software architect en oprichter van Solid Case B.V. Hij is te bereiken via armand.wijskamp@solidcase.nl.

geven. Het blijkt dat men soms direct denkt aan een set van applicaties die samen een midoffice vormen. Op zich is dat ook geen foute benadering. We hebben deze techniek tenslotte nodig om het een en ander te realiseren. Zelf geef ik de voorkeur aan een technologie onafhankelijke uitleg. De tijd heeft ons geleerd dat sommige technologie van tijdelijke aard is en dat applicaties drie tot vijf jaar mee gaan, nietwaar? Een SOA beschrijft een informatie architectuur als een verzameling van autonome services. Een webservice is een collectie van operaties waarvan het volgende bekend is: de functionaliteit, de pre- en postcondities en last but not least de interface. Deze services implementeren een afgebakend deel van, of een compleet bedrijfsproces. Hierbij is het streven naar een zo generiek mogelijk ontwerp met als doel een zo hoog mogelijke mate van herbruikbaarheid in verschillende bedrijfsprocessen. Een SOA is geen architectuur, maar een architectuurstijl.

Plan van aanpak

Hoe moet je in de hierboven geschetste situatie te werk gaan? Met een implementatie van een SOA is het in ieder geval mogelijk om diensten aan te bieden in de vorm van webservices. De vraag is alleen wel hoe je dat zou moeten doen. Van een SOA is bekend dat er eerst op korte termijn moet worden geïnvesteerd om op lange termijn voordelen te behalen. De opdrachtgever heeft aangegeven dat de gedane investeringen niet verloren mogen gaan. Daarnaast is er te weinig technische kennis op de afdeling aanwezig om op heel korte termijn met de tools van één van de SOA-marktleiders van start te gaan. Om deze redenen moet worden besloten om de implementatie van de bedrijfslogica daar te laten waar deze nu zit, namelijk in de backoffice legacy. Met andere woorden: de huidige bedrijfslogica moet als webservice beschikbaar worden gemaakt. Hiervoor wordt een integratielaag boven de legacy-laag geplaatst. De componenten in deze integratielaag zijn ontsluitbaar als webservice en zijn verantwoordelijk voor de interface en de communicatie met de legacy software component. Alle complexiteit met betrekking tot SOAP, HTML-headers en endpoints wordt uit de legacy-laag gehouden. Op deze manier hoeven bouwstandaarden van de verschillende backoffice applicaties niet te worden aangepast.

Analyse van bestaande software

Het ontsluiten van backoffice componenten als webservice klinkt mooi. De realiteit is dat veel van de bedrijfslogica hoogstwaarschijnlijk niet direct geschikt is om ook daadwerkelijk als webservice te worden ontsloten. Dit kan technische redenen hebben, maar er moet ook rekening mee worden gehouden dat de softwarecomponenten mogelijk niet atomair genoeg zijn om als webservice operatie te

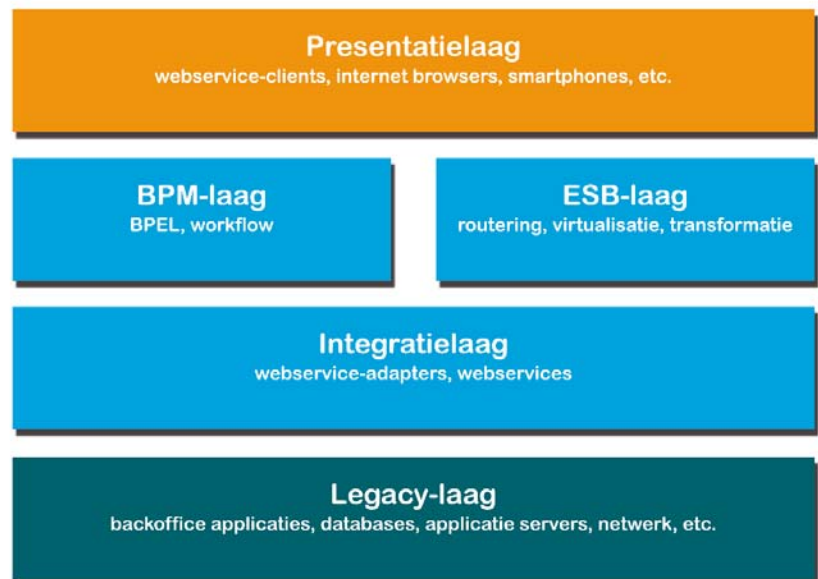
worden ontsloten. Dat zou kunnen betekenen dat de ontwerpers terug moeten naar de tekentafel. Om dit soort risico's op waarde in te schatten is een analyse nodig. Het is te tijdrovend om dit op voorhand voor alle geïmplementeerde functionaliteit te doen. Een pragmatische werkwijze verdient hier de voorkeur. Begin daarom pas met een analyse van de bestaande software op het moment dat functionaliteit ook daadwerkelijk als webservice moet worden ontsloten.

BPM en BPEL

De bedrijfsprocessen worden weer centraal gesteld. Om deze processen inzichtelijker te maken en het adaptief vermogen te verhogen moet een Business Process Modeling (BPM) laag worden geïntroduceerd. In deze laag kunnen nieuwe processen worden ontworpen die gebruikmaken van bestaande bedrijfsprocessen. Het te ontwerpen proces kan hierbij gebruikmaken van delen bedrijfslogica die zijn geïmplementeerd in verschillende applicaties. Door het toevoegen van deze nieuwe BPM-laag in de architectuur kunnen bedrijfsprocessen op één plek worden beheerd. Bedrijfsprocessen in de BPM-laag worden geïmplementeerd in de vorm van BPEL (Business Process Execution Language) scripts. BPEL is een op XML gebaseerde scripttaal waarmee bedrijfsprocessen kunnen worden opgebouwd uit aanroepen naar webservices. Tevens kunnen met BPEL ook taakgestuurde workflow processen worden gebouwd. De BPEL scripts zijn zelf ook aan te roepen als webservice. Doordat de BPM-laag zich bevindt boven de integratielaag is het ook mogelijk bedrijfsprocessen te modelleren die gebruikmaken van service-implementaties in bestaande applicaties. Je zou BPEL bijvoorbeeld kunnen inzetten voor een webservice die de prijs

Een SOA is geen architectuur, maar een architectuurstijl

Afbeelding 1. Lagen in de nieuwe architectuur.



Beginnen met het opdelen van de business in logische domeinen is een goede werkwijze

en levertermijn van een product opvraagt bij verschillende leveranciers. Zo'n webservice kan dan een generiek element teruggeven dat bestaat uit de naam van de leverancier, de prijs van het product en de levertermijn. Het is natuurlijk ook mogelijk om deze webservice alleen de gegevens te laten terug serveren van de leverancier met de laagste prijs of de kortste levertermijn. BPEL heeft een absolute meerwaarde in een omgeving waar bedrijfslogica meervoudig is geïmplementeerd in verschillende applicaties. Het is nu namelijk mogelijk om één implementatie leidend te maken en deze als webservice te publiceren.

Wat is BPEL?

BPEL is een op XML gebaseerde scripttaal waarmee bedrijfsprocessen kunnen worden opgebouwd uit aanroepen naar webservices. Tevens kunnen met BPEL ook taakgestuurde workflow processen worden gebouwd. De BPEL scripts zijn zelf ook aan te roepen als webservice.

Webservice ontwerp

Heel leuk al die webservices, maar het is niet eenvoudig om een goede webservice (operatie) te ontwerpen. Zeker niet wanneer deze webservice atomair, persistent en generiek moet zijn. Een generieke interface is een stuk lastiger te ontwerpen dan een specifieke interface. Ook moet worden bepaald wat de scope van een webservice is. Hoe groot moeten de webservices zijn, of anders gezegd hoeveel operaties mag een webservice bevatten? Een goede werkwijze is beginnen met het opdelen van de business in logische gegevensdomeinen. In de praktijk hebben we te maken met allerlei bestaande systemen, maar voor een goede indeling van domeinen is het van groot belang om de indeling uitsluitend te maken op functionele basis. Kijk in dit traject dus niet naar de applicaties waarin de huidige business is geïmplementeerd. Een goede domeinindeling maakt de business transparanter aan de buitenkant. Voorbeelden van logische domeinen zijn Relatie, Product en Order. Indien het domein Relatie te groot zou worden kun je ervoor kiezen om het op te delen in de domeinen Klant, Leverancier en Werknemer. De domeinen dienen als basis voor het opstellen van het gegevensmodel. Een gegevensmodel van een domein bevat alle berichtelementen waarvoor dat specifieke domein verantwoordelijk is. Wanneer een relatie binnen de organisatie altijd wordt geïdentificeerd aan de hand van een relatienummer, relatietype en landcode, moet er een standaard element RelatieId worden gemodelleerd binnen het domein Relatie. Webservice operaties moeten dan altijd gebruikmaken van deze definitie wanneer een relatie moet worden opgenomen in de interface ervan.

Compromissen sluiten

Het ontwerpen van atomaire en herbruikbare webservice operaties en berichtelementen blijkt lastig in de praktijk. Het is wat kort door de bocht te denken dat de herbruikbaarheid maximaal is wanneer je zoveel mogelijk data rondom een domeinobject terugserveert. Dit kan als risico hebben dat queries in de database te veel tijd kosten en de XML berichten te groot. In dat geval dient een compromis te worden gesloten tussen performance en herbruikbaarheid. Het bepalen van een optimum hierin is simpelweg niet eenvoudig. Neem bijvoorbeeld een functie die de orderhistorie van een product toont. Hierbij moet per order ook de contactgegevens van de klanten zichtbaar zijn. Is het dan zinnig één webservice operatie te ontwerpen die al deze gegevens terug geeft? Je weet in dit geval niet zeker hoe groot te de terug te serveren XML wordt. Tevens voel je wel aan dat zo'n webservice operatie een lage herbruikbaarheidsgraad heeft. Het is in dit geval slimmer hiervoor verschillende operaties te ontwerpen. Bijvoorbeeld één webservice voor het ophalen van de productgegevens, een tweede voor de orderhistorie en een derde voor het teruggeven van de adresgegevens van de klanten.

Ontkoppelen

Door de inzet van een Enterprise Service Bus (ESB) wordt een stuk ontkoppeling tussen externe interface en implementatie gerealiseerd. ESB Routing Services roepen de implementatie services aan. Door alleen de routing services extern te publiceren is het mogelijk om wijzigingen in implementatie componenten aan te brengen zonder dat externe partijen hier iets van merken. Op deze wijze heb je de beschikking over een krachtig middel voor versiebeheer. Ook is hierdoor uitfaseren van software componenten een stuk eenvoudiger. In de ESB-laag kunnen XSLT (Extensible Stylesheet Language Transformations) transformaties worden uitgevoerd. Dit maakt het mogelijk aan de buitenkant een gestandaardiseerde interface te publiceren, terwijl de implementatieservice een specifieke interface kent. Dit is bijvoorbeeld handig voor organisaties die Nederlandstalige en Engelstalige applicaties hebben, maar alleen Engels als standaard hanteren voor extern gepubliceerde dienstverlening.

SOA-regie

Zonder een team dat verantwoordelijk is voor het bewaken van de architectuur heeft een implementatie van een SOA eigenlijk geen echte kans van slagen. Het SOA-regieteam doet reviews van de ontwerpen voor webservices en workflow processen. Hierbij moet in het begin veel aandacht worden besteed aan de functionele afbakening van de webservice operaties en de interfaces van deze operaties. De mensen van dit team weten tenslotte welke webservice operaties al gebouwd zijn, en welke

standaard berichtelementen al aanwezig zijn in het gegevensmodel. Zonder regie bestaat het risico dat bedrijfslogica alsnog meervoudig wordt gebouwd. Deze valkuil moet je voorkomen. Het verdient de voorkeur om tijdens de eerste SOA projecten het beheer van het gegevensmodel te laten doen door het regieteam.

Standaarden: lusten en lasten

Tijdens ontwerp- en bouwfasen van de eerste projecten is eigenlijk direct behoefte aan standaarden en richtlijnen. Het is echter praktisch onmogelijk om vooraf alles te voorzien. Vooraf opgestelde theoretische standaarden moeten nu eenmaal tijdens deze eerste projecten worden bijgesteld. Het gevolg hiervan is dat, met name in het begin, functionele ontwerpen en gebouwde componenten met terugwerkende kracht moeten worden aangepast. Wanneer je dit niet gelijk doet leidt dit tot een situatie met veel versies van interfaces en componenten. Niemand zit te wachten op een negatief effect op onderhoudbaarheid en beheersbaarheid dus je hebt eigenlijk geen keus. Dit kan frustratie opwekken, zeker wanneer er een druk op het project wordt uitgeoefend om snel iets op te leveren. Het zorgt in ieder geval voor een langere doorlooptijd, maar daar is natuurlijk een factor voor in te plannen.

Alle begin...

Het risico bestaat dat de eerste projecten veel langer duren dan vooraf ingeschat. Dit is met name te wijten aan de relatief hoge investeringen die voor de korte termijn moeten worden gedaan. Voor alle

betrokkenen in het project geldt dat er iets nieuws om de hoek kwam kijken. De grootste hobbel blijkt de andere manier van denken te zijn en niet de techniek zoals vooraf werd gedacht. Tijdens de eerste projecten moet extra tijd worden ingepland voor zaken als:

- bewaking architectuur;
- standaarden en richtlijnen;
- operationeel beheer;
- opleiding;
- configuratiebeheer;
- versiebeheer;
- Software Level Agreements (SLA's).

Toekomstige baten

De toevoeging van een integratielaag heeft ertoe geleid dat externe dienstverlening in de vorm van webservices aan de buitenwereld kan worden aangeboden. Daarnaast is het eenvoudiger geworden om met webservices van derden te communiceren. Verder staat er met de toevoeging van de BPM-laag een raamwerk dat kan worden ingezet voor implementatie van taakgeoriënteerde workflow. De meeste voordelen zullen echter pas op langere termijn zichtbaar worden. Hierbij moet je denken aan verlenging van de levensduur van de bestaande systemen en kostenbesparing door verhoogd hergebruik van software. Tevens wordt het ontwikkelproces versneld, wordt de beheersomgeving flexibeler en kunnen bedrijfsprocessen worden geoptimaliseerd door invoering van workflow. Alles bij elkaar opgeteld goede redenen om legacy te recyclen met behulp van een SOA. «

Recyclen van legacy met een SOA werpt in de toekomst zijn vruchten af

Patches Patches Patches Patches Patches Patches Patches Patches

Artikelen over onderwerpen als software-ontwikkeling, Java, UML, eXtreme Programming en nog veel meer vindt u in het Online Archief van Array Publications. Vaktijdschriften als Software Release, Java Magazine, Database Magazine en ons Oracle vakblad Optimize hebben hun artikelenarchief online gezet. Dankzij de heldere zoekstructuur vindt u snel wat u zoekt op www.release.nl.

Nitro PDF wil Europese markt veroveren

Nitro PDF heeft onlangs een Nederlandse versie van haar PDF-conversieprogramma uitgebracht. Het Amerikaanse bedrijf presenteert zich als een alternatief voor Adobe Acrobat en Nuance PDF Converter en wil de Europese markt veroveren.

Het Amerikaanse bedrijf kwam eerder dit jaar al met een Duitstalige versie en richt zich na de Nederlandse presentatie van haar producten ook op Frankrijk, Spanje en Italië. Sam Chandler, CEO van Nitro PDF, maakt een rondreis door Europa om zijn product aan de man te brengen. In het Engelse taalgebied is Nitro PDF

al een stevige partij op de markt van PDF-conversie. "Er zijn al miljoenen mensen die onze programma's gebruiken. Onze websites voor vrije conversie van PDF naar Word en Excel kennen al meer dan dertig miljoen bezoekers en daar komen er maandelijks een miljoen bij. Dat willen we ook in Europa bereiken", verklaart hij.

Chandler claimt dat zijn programma sneller en gebruiksvriendelijker is dan dat van zijn concurrenten. De Australiër heeft daarom ook veel vertrouwen in zijn product.

"Mensen kunnen via onze websites gratis alles proberen. De conversie van documenten is eenvoudig. Na het invoeren via één van de websites, krijg je bin-

nen een paar minuten de omgezette versie van of naar PDF terug per email. De eenvoud daarvan moet zowel zakelijke klanten als consumenten over de streep trekken om een pakket aan te schaffen. En dat is goedkoper dan onze concurrenten", meent hij.

Microsoft-interface

Daarnaast roemt Chandler ook de interface van zijn product. Een korte demonstratie laat zien dat die erg veel weg heeft van het uiterlijk van Microsoft Office. "Daarvoor hebben we gekozen, omdat mensen daar vertrouwd mee zijn. Ze hoeven niet te wennen aan een nieuw uiterlijk om ons product te gebruiken."

De conversie van PDF naar Word (of omgekeerd) is volgens Chandler zo nauwkeurig, dat je niet eens van conversie kunt spreken. Nitro PDF noemt de hiervoor ingezette techniek 'intelligent assessment'.

De software bestudeert als het ware het Word-document en bepaalt welke functies bepaalde onderdelen hebben: kolommen, tabellen en graphics worden als zodanig herkend en vertaald naar de nieuwe documentvorm. Dit betekent dat een vanuit PDF naar Word geconverteerde tabel daadwerkelijk als tabel benaderbaar blijft.

Voor meer informatie:
<http://www.nitropdf.com/nl>