

GEGEVENSUITWISSELING (INTERFACING) VAN APPLICATIES MET BEHULP VAN EEN MESSAGE BROKER

DE APPLICATIES VAN DE BUSINESS UNIT BEDRIJFSCOMMUNICATIE VAN KPN TELECOM

Bedrijfsprocessen worden meer en meer ondersteund door informatiesystemen, die onderdeel uitmaken van de ICT. Vragen die men zich vaak stelt zijn: "Hoeveel en welke informatiesystemen heeft het bedrijf / de organisatie nodig?" en "Welke gegevens moeten er tussen de informatiesystemen worden uitgewisseld?". Vaak vindt gegevensuitwisseling plaats met behulp van zogenoemde 1 op 1 koppelingen/interfaces. Gegevens van het ene informatiesysteem worden aangeboden aan het andere informatiesysteem en vice versa. Organisaties met honderden informatiesystemen hebben vaak ook een groot aantal interfaces. Ook bij de Business Unit Bedrijfs-Communicatie (BU BC) van KPN Telecom is dat het geval. Het onderhoud van alle interfaces vergt een omvangrijke personele en financiële inspanning. Om het beheer van de interfaces beter beheersbaar te maken en met name de kosten sterk omlaag te brengen heeft de BU BC twee jaar geleden besloten tot aanschaf van de message broker ENGIN. Met behulp van de message broker wordt een en ander bereikt. Dit artikel gaat over de wijze waarop de message broker wordt ingezet en de ervaringen daarmee. Zeer recent is ook Baan ERP gekoppeld aan de message broker. Op die manier vindt gegevensuitwisseling tussen de bedrijfsapplicaties en Baan ERP plaats.

1 Inleiding

1.1 ONDERWERP VAN DIT ARTIKEL

Bedrijfsprocessen worden meer en meer ondersteund met behulp van informatiesystemen, die onderdeel uit-

maken van de ICT. Vragen die men zich vaak stelt zijn: "Hoeveel en welke informatiesystemen heeft het bedrijf/ de organisatie nodig?" en "Welke relaties bestaan er tussen de informatiesystemen en welke gegevens moeten er tussen de informatiesystemen worden uitgewisseld?".

De afgelopen decennia was over deze vragen veel dis-

cussie. Tussen twee uitersten ontstonden veel tussenvormen. De uitersten waren enerzijds een groot aantal losse informatiesystemen die niet of nauwelijks met elkaar communiceerden en anderzijds het streven naar één allesomvattend informatiesysteem. De eerste vorm kan worden gekwalificeerd als "eilandautomatisering". Dit kwam in de zestiger en zeventiger jaren veel voor, maar ook heden ten dage is een dergelijke vorm van informatisering hier en daar nog waarneembaar. De tweede vorm was populair in de jaren zeventig. Veel informatici, met name zij die weinig daadwerkelijke praktijkervaring hadden, dachten met behulp van één geïntegreerd en allesomvattend, informatiesysteem alle informatieproblemen te kunnen oplossen. Dat ging echter volledig mis. De beoogde omvangrijke monolithische informatiesystemen kwamen niet van de grond.

De juiste oplossing is een beperkt aantal informatiesystemen. In hoofdstuk 2 wordt hierop nader ingegaan. Belangrijk daarbij is het uitwisselen van gegevens tussen de informatiesystemen. Vaak zijn dat zogenoemde 1 op 1 koppelingen/interfaces (zie figuur 3). Gegevens van het ene informatiesysteem worden aangeboden aan het andere informatiesysteem en vice versa. Organisaties met honderden informatiesystemen hebben vaak ook een groot aantal interfaces. Ook bij de Business Unit BedrijfsCommunicatie (BU BC) van KPN Telecom is dat het geval. Het onderhoud van alle interfaces vergt een omvangrijke personele en financiële inspanning. Om het beheer van de interfaces beter beheersbaar te maken en met name de kosten sterk omlaag te brengen heeft de BU BC twee jaar geleden de message broker ENGIN aangeschaft. Met behulp van de message broker wordt een en ander bereikt. Dit artikel gaat over de wijze waarop de message broker wordt ingezet en de ervaringen daarmee. Zeer recent is ook Baan ERP gekoppeld aan ENGIN en vindt gegevensuitwisseling tussen de bedrijfsapplicaties en Baan ERP plaats.

Dit hoofdstuk beschrijft de positie van de BU BC binnen KPN Telecom. In hoofdstuk 2 komt de probleemstelling aan de orde. Een korte schets van twee informatiesystemen wordt gegeven in hoofdstuk 3, terwijl het koppelen van applicaties uitgebreid aan de orde komt in hoofdstuk 4. De conclusie en toekomstige ontwikkelingen besluiten het artikel.

1.2 DE POSITIE VAN DE BU BEDRIJFSCOMMUNICATIE BINNEN KPN TELECOM

KPN Telecom is in Nederland een zeer belangrijke speler op de telecommunicatiemarkt en moet snel en flexibel inspelen op de dynamiek in de telecommunicatiemarkt en de steeds veranderende wensen van de klanten. KPN doet dat door een gedifferentieerd producten- en dienstenaanbod. Daartoe is de organisatie van KPN ingedeeld in Business Units, Verkoop-eenheden en Productie Eenheden. Deze eenheden worden ondersteund door Staven, Professionele Service Eenheden en Gemeenschappelijke Service Eenheden. Met deze indeling wordt gestreefd naar een strakke sturing van businessdoelstellingen op verlies en winst en een heldere en eenduidige toekenning van verantwoordelijkheden en bevoegdheden.

Productie-eenheden zijn onder andere: het *Software Huis*, dat diensten levert aan de KPN eenheden op het gebied van applicatiesoftware (kennis, ontwikkelen, integreren, testen en beheer) en het *Datacenter*, dat diensten levert voor de implementatie, exploitatie en beheer van ICT-infrastructuur. Professionele Service Eenheden zijn onder andere: *Innovatie Unit ICT* (is gericht op innovatie van klantprocessen en leveringsprocessen), *Audit* (levert informatie en advies over de beheersing van de bedrijfsprocessen) en *KPN Research* (onderzoek en ontwikkeling op het gebied van technologie en sociale wetenschappen).

KPN Telecom omvat de volgende Business Units:

- Business Unit Carrier Services;
- Business Unit Vaste Telefonie;
- Business Unit Mobiele Telefonie;
- Business Unit Corporate Networks;
- Business Unit TeleCommerce;
- Business Unit BedrijfsCommunicatie.

1.3 DE BUSINESS UNIT BEDRIJFSCOMMUNICATIE

De BU BC houdt zich bezig met de inkoop, verkoop, installatie en service van alle apparatuur en systemen die bij klanten worden geplaatst. Daarnaast zorgt de BU BC voor het maken van aansluitingen in het aansluitnet. In het werkveld van de BU BC zijn zo'n 6000 mensen werkzaam: veel monteurs, planners en werkvoorbereiders maar ook marketeers, financiële medewerkers en personeelsfunctionarissen. Het doel van de BU BC is marktlei-

der zijn en blijven en het realiseren van een goed rendement. Om dat doel te bereiken werkt de BU continu aan het verbeteren van de bedrijfsprocessen. Belangrijke zaken daarbij zijn het neerzetten en onderhouden van een heldere, effectieve en efficiënte organisatie en het bevorderen van het nemen van initiatief en verantwoordelijkheid van medewerkers naar klanten. De bedrijfsprocessen van de BU BC worden ondersteund door ICT. Dit is een taak van de Business Proces Group.

De BU BC bestaat uit de volgende bedrijfsonderdelen (zie figuur 1):

- Business Line Bovenband;
- Business Line Keysystems;
- Business Line Cablecom;
- Business Line Aansluitnet;
- Business Line Communication Services;
- Business Proces Group;
- Communicatie en Kwaliteit;
- Business Support;
- Personeelszaken;
- Financiën.

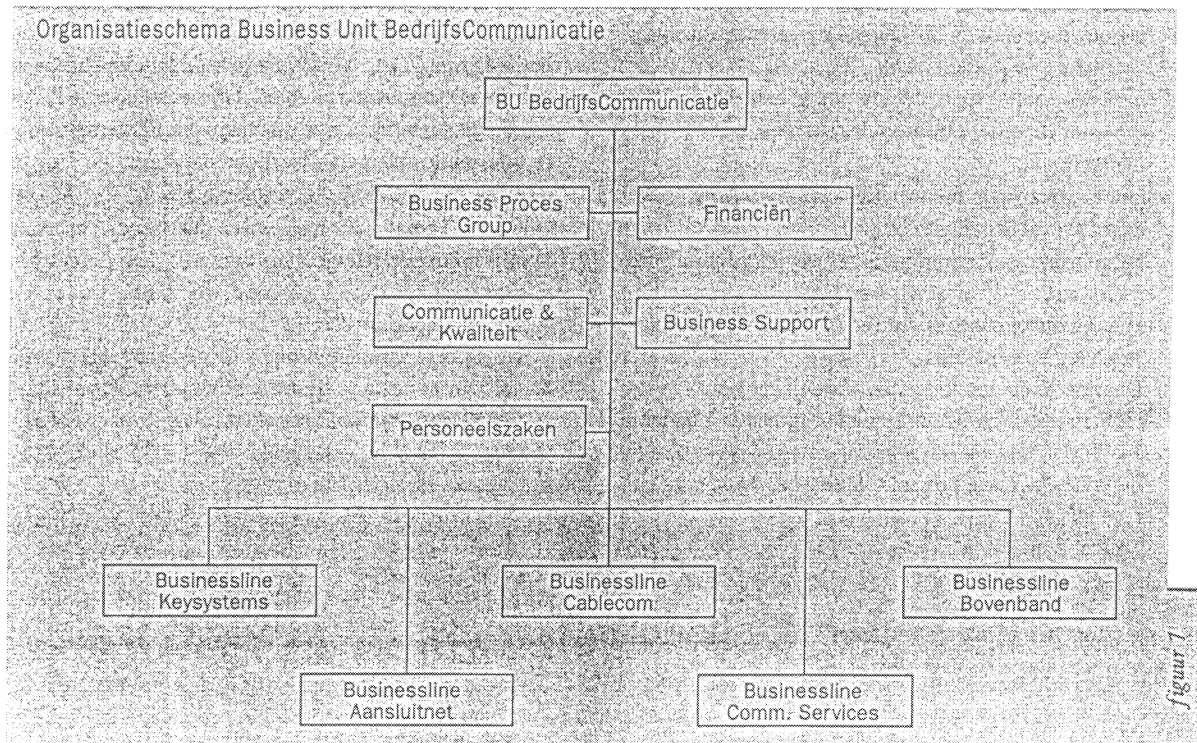
2 Probleemstelling

2.1 INLEIDING

In hoofdstuk 1 is aangegeven dat noch eilandautomatisering noch één allesomvattend informatiesysteem op adequate wijze de bedrijfsprocessen van een organisatie kunnen ondersteunen. En daar gaat het toch om, dat is de toegevoegde waarde van ICT.

De praktijkervaringen van de auteurs van dit artikel sluiten nauw aan bij de visie van prof. Nielen [Nielen 1989]. Deze stelt onder andere:

- grote informatiesystemen moeten niet als één geheel worden ontworpen;
- als we de omvang van een informatiesysteem vergroten neemt de moeilijkheidsgraad exponentieel toe;
- het is verstandig om verschillende systemen te hantieren, die ieder een specifiek stel eigenschappen hebben;
- grote informatiesystemen mogen niet meer lijken op een kantoorgebouw dat ontworpen, gebouwd en betrokken wordt; ze moeten daarentegen lijken op het personeel van de organisatie dat verloopt en wordt aangevuld;
- voor het onderkennen van de optimale subsystemen is



- een grondige kennis vereist van de onderneming waarvan het informatiesysteem de besturing ondersteunt;
- samenvattend: informatiesystemen moeten niet te groot zijn, niet teveel functionaliteiten bezitten en moeten weggegooid en vervangen kunnen worden wanneer ze niet meer voldoen.

In 1998 heeft de BU BC, geconfronteerd met haar verzameling informatiesystemen en interfaces, geconcludeerd (probleemstellingen) dat:

- sommige informatiesystemen teveel functionaliteiten bezitten;
- sommige informatiesystemen een monolithisch karakter hebben;
- er erg veel 1 op 1 interfaces zijn;
- het daardoor moeilijk wordt om informatiesystemen te vervangen;
- het moeilijk is om snel in te spelen op veranderingen in de business;
- het onderhoud van de informatiesystemen en interfaces zeer hoge kosten met zich meebrengt.

Dat heeft geleid tot een aantal belangrijke beslissingen (doelstellingen):

- informatiesystemen voor de BU BC moeten worden gebouwd volgens de drie-lagen-architectuur: data laag, functielaag en presentatielaag [Stee 1997];
- per laag worden zogenoemde bouwblokken gemaakt, die door meerdere informatiesystemen kunnen worden gebruikt;
- het bouwen van nieuwe informatiesystemen moet daardoor sneller en goedkoper;
- hergebruik van componenten wordt gestimuleerd (beloond);
- een uitdaging is het realiseren van een adequate verzameling bouwblokken.

Daartoe zou gebruik kunnen worden gemaakt van het "Biljartmodel" (zie figuur 2). Een beginnende biljarter is

blij als hij de stootbal goed raakt en op die manier een carambole maakt. Een professionele biljarter wil niet alleen de carambole maken, maar ook iets goeds (een goede ligging van de ballen) overhouden. Van een professionele informaticus mag men iets dergelijks verwachten. Een professionele projectleider scoort niet alleen (opleveren van afgesproken functionaliteiten, op tijd, binnen het budget, voldoen aan kwaliteitseisen) maar houdt ook iets goeds (bouwblokken waar de organisatie iets aan heeft) over. Dit is een interessant beoordelingsaspect;

- de verzameling interfaces moet worden gesaneerd. Dit onderwerp staat centraal in dit artikel.

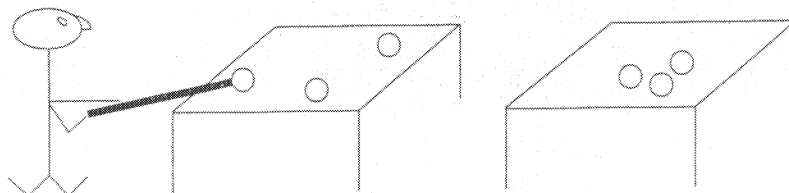
2.2 PROBLEEM

Gegevensuitwisseling tussen informatiesystemen kan op verschillende manieren.

Vanouds is er de 1 op 1 koppeling/interface. Figuur 3 schetst zo'n interface tussen twee informatiesystemen van de Technische Universiteit Delft die op die manier hebben gefunctioneerd in de periode 1985-1999 [Dijk84]. Het voorraadbeheer- en magazijnsysteem GEMS ontving iedere nacht per magazijn de geldige klantnummers. Het financieel systeem FINS ontving periodiek de facturingsgegevens. Deze 1 op 1 interface (in feite zijn het twee interfaces) is gemaakt in 1985. Hij beviel zo goed, dat soortgelijke interfaces zijn gemaakt voor FINS met andere informatiesystemen. Daardoor ontstond een 1 op n interface (zie figuur 4). Al filosoferend kwam de wens op tafel om te komen tot een algemeen interfacemechanisme voor n op m koppelingen (zie figuur 5). Een dergelijke, algemeen toepasbare, oplossing viel echter buiten de scope van de TU Delft.

De huidige message brokers bieden dergelijke faciliteiten. Dit was voor de Business Proces Group aanleiding om in 1998 tot aanschaf van de message broker ENGIN

Biljartmodel (raken en iets goeds overhouden)



figuur 2

DE MENING VAN ... MONIQUE CLAUS

Monique Claus is werkzaam bij de Business Proces Group van de Business Unit BedrijfsCommunicatie. Zij is betrokken bij architectuurvraagstukken. Monique: "Wij richten ons bij applicatieontwikkeling sterk op de drie-lagen-architectuur: presentatielaag, functielaag en gegevenslaag. Nieuwe applicaties, zoals CAPS dat wordt gebruikt voor de afhandeling van capaciteitsaanvragen, worden volgens deze architectuur opgezet. Bestaande applicaties worden bij groot onderhoud "uit elkaar getrokken". Naast deze drie-lagen-architectuur werken wij steeds meer met componenten of bouwblokken. Deze kunnen door meerdere applicaties worden gebruikt. Op die manier bestaat een applicatie voor een belangrijk deel uit een verzameling bouwblokken met een besturings-element. De vraag is hoe bouwblokken met elkaar communiceren. Soms communiceren de bouwblokken

via de message broker. Dat werkt prima. Als de bouwblokken klein zijn, worden ze vaak geassembleerd tot een groter bouwblok. Als we alle kleine bouwblokkjes via de message broker zouden laten communiceren, dan zou dat erg veel "overhead" geven en in sommige gevallen vertragend werken. Toch is de omvang van een bouwblok niet altijd bepalend. Zo hebben we een bouwblok voor assortimentsgegevens dat niet zo groot is, maar dat wel door allerlei applicaties wordt geraadpleegd. De message broker speelt daarbij een essentiële rol.

De keuze of bouwblokken worden samengevoegd tot een groter bouwblok of dat ze communiceren via de message broker moet iedere keer zorgvuldig worden afgewogen!".

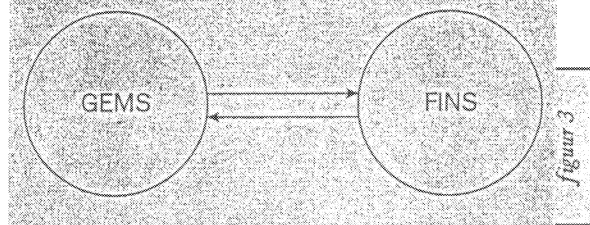
over te gaan.

De probleemstelling:

"Er zijn teveel 1 op 1 interfaces en het onderhoud van de interfaces brengt te hoge kosten met zich mee" kon als volgt worden vertaald naar een oplossing:

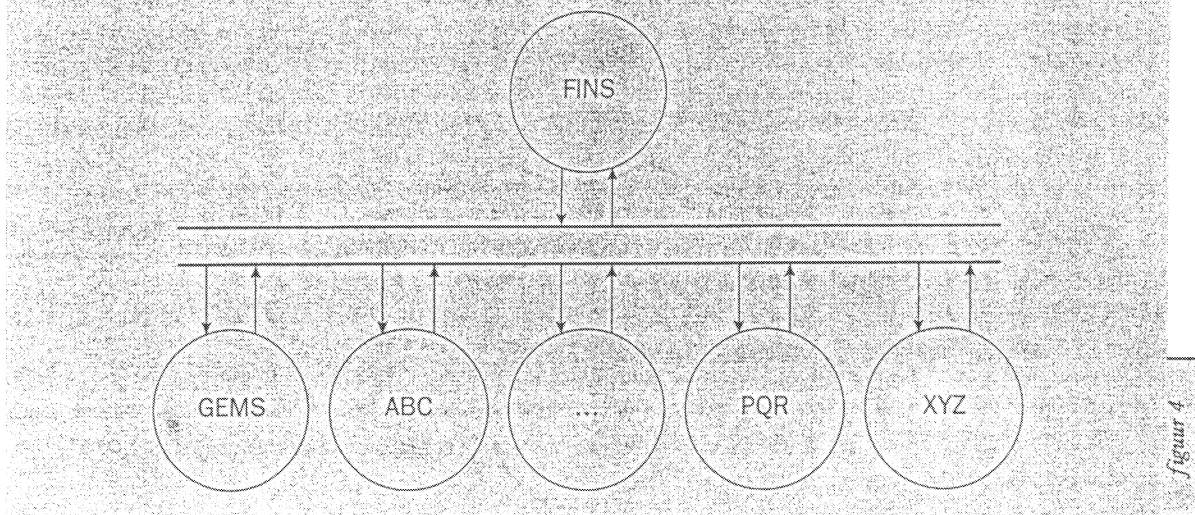
- realiseer zoveel mogelijk nieuwe interfaces via de message broker;
- vervang bij groot onderhoud van de huidige interfaces deze door interfaces via de message broker, tenzij er door het management goed te keuren redenen zijn om

Gegevensuitwisseling tussen twee applicaties (1:1)



figuur 3

Gegevensuitwisseling tussen een applicatie en een aantal andere applicaties (1: n)



figuur 4

- dat niet te doen;
- rapporteer over de voor- en nadelen van het gebruik van de message broker.

ook in het volgende hoofdstuk aan de orde.

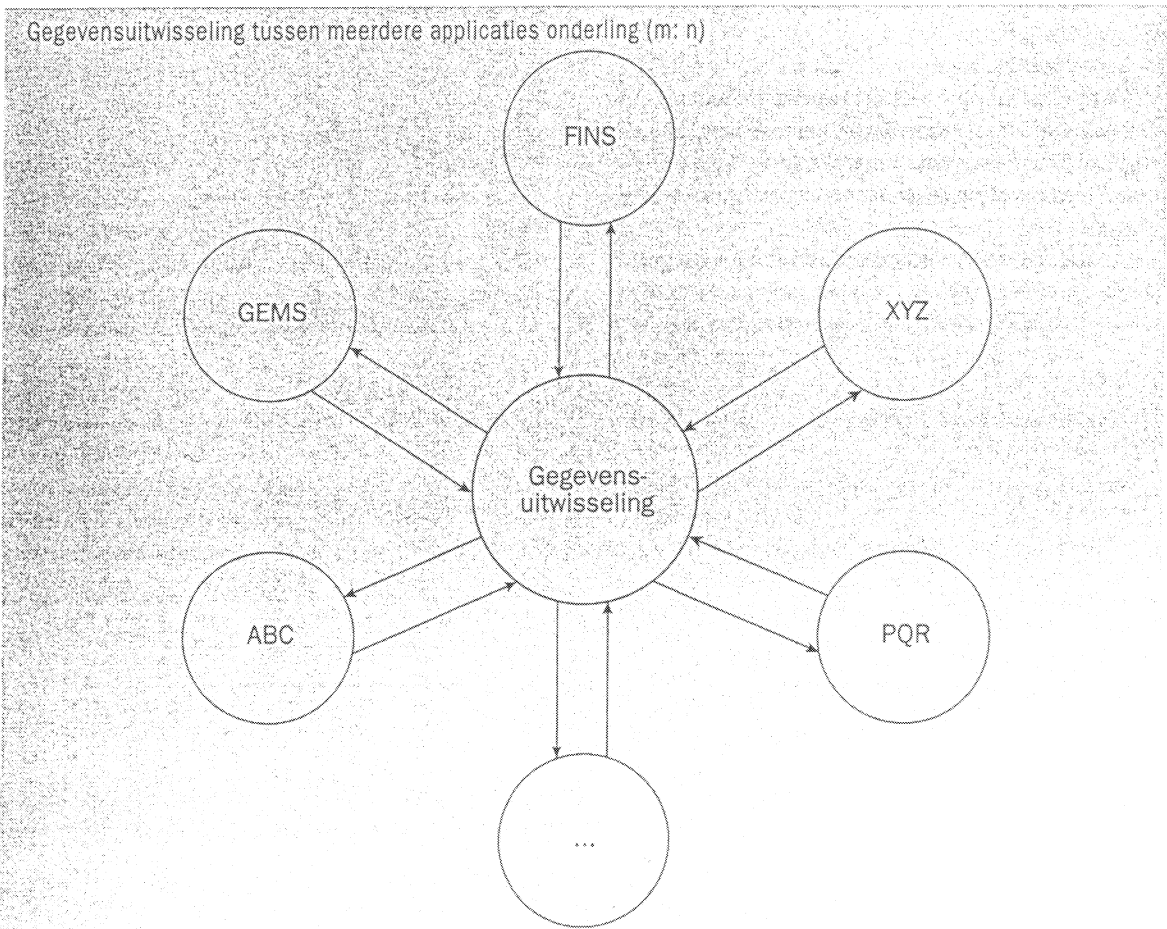
3 Casus

3.1 INLEIDING

Zoals in de vorige hoofdstukken is aangegeven, heeft de BU BC een groot aantal informatiesystemen, waaronder een aantal "landelijke systemen" [Clau1999]. Deze informatiesystemen hebben vaak onderlinge koppelingen/interfaces ten behoeve van gegevensuitwisseling. In dit hoofdstuk worden in het kort twee, voor de BU BC zeer belangrijke, informatiesystemen alsmede een interface tussen deze informatiesystemen besproken. De betreffende informatiesystemen en de interface komen

3.2 M&M007

Het informatiesysteem M&M007 (Meld- en Meetpost 007) wordt gebruikt voor de registratie en afhandeling van storingsmeldingen betreffende telecommunicatieapparatuur en -installaties. Met behulp van M&M007 worden storingsmeldingen geregistreerd, geanalyseerd en afgehandeld. Het informatiesysteem biedt ondersteuning bij het vertalen van de storing naar een serviceorder en gebruikt, indien beschikbaar, daarbij het betreffende servicecontract.



figuur 5

Het informatiesysteem ondersteunt onder andere de volgende processen:

- het aannemen van klachtmeldingen;
- het analyseren van klachten;
- het verstrekken van serviceorders aan monteurs om een storing op te heffen;
- bewaken van de voortgang van de afhandeling van een klachtmelding;
- verstrekken van informatie over de klachtmeldingen;
- gereed melden aan de klant van de klachtafhandeling.

M&M007 heeft interfaces met acht informatiesystemen, waaronder FIT.

3.3 FIT

Met behulp van het informatiesysteem FIT (Field Information Terminal) worden monteurs elektronisch aangestuurd. Daartoe beschikken de monteurs over een mobiele werkplek en een GSM toestel. FIT bestaat uit twee deelsystemen: FIT Centraal en FIT Mobiel. FIT Centraal wordt gebruikt voor toewijzing, planning, bewaking van klantorders en het vervaardigen van managementinformatie. FIT Mobiel is het deel van het FIT informatiesysteem dat op de PC's van de monteurs is geïnstalleerd. De monteurs gebruiken FIT Mobiel voor de afhandeling van klantorders. Het informatiesysteem ondersteunt onder andere de volgende processen:

- in behandeling nemen van serviceorders;
- toewijzen van serviceorders;
- bewaken van de voortgang van serviceorders;
- verwerken van materiaal aanvragen;
- bestellen van artikelen;
- afmelden uitgevoerde werkzaamheden bij de klant;
- gereed melden van de serviceorder.

FIT heeft interfaces met zes informatiesystemen, waaronder M&M007.

3.4 INTERFACES M&M007 EN FIT

Tussen M&M007 en FIT bestaan de volgende interfaces:

- M&M007 biedt serviceorders aan aan FIT die deze in ontvangst neemt;

- FIT informeert M&M007 over de voortgang van de serviceorders en biedt terug- en gereedmeldingen aan.

Tabel 1 geeft een overzicht van de betreffende transacties.

4 Koppeling van applicaties

4.1 DE ROL VAN DE MESSAGE BROKER

Enterprise Application Integration (EAI) is het proces dat de machines van een of meer bedrijven via een netwerk met elkaar verbindt [Rock2000], zodat zowel nieuwe applicaties als legacy-applicaties en hun gegevensbestanden kunnen worden geïntegreerd. De integratie vereist daarbij weinig of geen wijzigingen van de bestaande applicaties. Een belangrijke EAI-component is de message broker. Gartner [Dec1998] geeft als definitie en doel van een message broker:

"Message brokers are logical hubs that copy and forward messages to one or more destinations. A message broker is an intelligent third party (hence a 'broker') between information sources and information consumers. The purpose of this architectural type is to integrate independently designed application domains (e.g., new applications, purchased applications and legacy applications). Message brokers make it possible to re-engineer business processes without re-engineering all the application programs and databases."

Een message broker is een (centraal gepositioneerde) middleware applicatie die het mogelijk maakt om andere applicaties eenvoudiger met elkaar te laten communiceren door het uitwisselen van gegevens. Er kunnen drie vormen van gegevensuitwisseling worden onderscheiden [Over1998], te weten:

- *Een vraag om gegevens:* applicatie A stelt een vraag aan applicatie B. Deze stuurt het antwoord op de vraag of een afwijzing (voorbeeld: "niet geautoriseerd voor deze gegevens");
- *Een verzoek tot wijziging van gegevens:* applicatie A wil dat applicatie B een of meer gegevens wijzigt. Dit kan een inhoudelijke wijziging zijn (voorbeeld: de mededeling dat een adres is gewijzigd) of het toevoegen of verwijderen van een gegeven (voorbeeld: toevoegen van een adres);
- *Een opdracht tot wijziging:* applicatie A meldt aan applicatie B dat er iets van zijn gegevens moet worden

Tabel 1: Overzicht interfaces M&M007 waaronder die met FIT

M&M007 transacties		FIT transacties
<i>Serviceorder verwerken door externe systemen:</i>		
Leveren van een serviceorder	ext-ord-1	Ontvangen van een serviceorder
Ontvangen van een terugmelding	ext-ord-2	Leveren van een terugmelding
Ontvangen van voortgangsinformatie	ext-ord-3	Leveren van voortgangsinformatie
Ontvangen van een gereedmelding	ext-ord-4	Leveren van een gereedmelding
<i>Serviceorder verwerken door M&M007:</i>		
Ontvangen van een serviceorder	int-ord-1	
Leveren van voortgangsinformatie	int-ord-2	
Leveren van een gereedmelding	int-ord-3	
Leveren van informatie over de serviceorder	int-ord-4	

gewijzigd (voorbeeld: het wijzigen van een projectstatus). Deze laatste vorm doet zich voor wanneer applicatie A de "eigenaar" is van bij applicatie B opgeslagen gegevens.

Applicaties hebben vaak een verschillend datamodel. Daarom is het wenselijk om tijdens het uitwisselen van gegevens de mogelijkheid te hebben om gegevens te transformeren. Onder transformeren wordt in dit geval zowel *converteren* (veldinhoudelijke wijzigingen, bijvoorbeeld het omzetten van een datum) als *mappen* (het omzetten van recordlayouts) verstaan. Daarmee is de communicatie op applicatieniveau geregeld. Naast deze communicatielaag is er de communicatie op transportniveau. Daarin wordt beschreven hoe de zender en ontvanger samenwerken. De twee lagen van communicatie zijn in principe onafhankelijk van elkaar.

Samenvattend kan worden gesteld dat een message broker een snelle postbode is (gegevensuitwisseling) die, alvorens de gegevens te bezorgen, de gegevens op een door de ontvanger gewenste manier omzet (transformeren) en daarmee een uitstekend communicatiemiddel is om applicaties met elkaar te laten praten.

De BU BC heeft in 1998 de message broker ENGIN van de Amerikaanse firma Muscato aangeschaft. Het uitwisselen en transformeren kan worden toegelicht aan de hand van figuur 6.

Iedere aan de message broker aangesloten applicatie verleent een aantal diensten met betrekking tot gegevensuitwisseling. Daartoe wordt van iedere dienst een zogenoemde Service SPecificatie (SSP) gemaakt. Dus applicatie A zegt als het ware: hier is mijn dienst A3 en wie de dienst wil en mag afnemen, vraagt dat maar aan de message broker. Het zal applicatie A, tot op zekere hoogte, een zorg zijn welke applicaties allemaal dienst A3 afnemen. Indien nu applicatie A volgens dienst A3 (voorbeeld: verzenden van projectgegevens) gegevens wil verzenden naar applicatie B met dienst B5 (voorbeeld: ontvangen van projectgegevens), dan zal de message broker de betreffende gegevens van applicatie A naar applicatie B sturen. Alvorens de gegevens aan applicatie B aan te bieden zal de message broker de gegevens transformeren. De wijze waarop dit dient te geschieden is binnen de message broker vastgelegd met behulp van een Service Implementatie SPecificatie (SISP).

Zonder te streven naar volledigheid wordt nu op een aantal aspecten van het werken met een message broker nader ingegaan.

Interface-awareness

De message broker kan onder andere worden gebruikt om gegevensuitwisseling tussen bestaande legacy-systemen te realiseren. Dat neemt niet weg dat moet worden gestreefd naar "interface-awareness" van applicaties. Dat betekent dat applicaties er rekening mee houden dat ook de buitenwereld de gegevens en de functionaliteit van de applicatie wil gebruiken. Omgekeerd moeten applicaties er rekening mee houden dat sommige gegevens uit de buitenwereld komen. Door een goede "Interface-awareness" van de op de message broker aangesloten applicaties wordt niet alleen de rol van de message broker beperkt, maar kan ook sneller worden ingespeeld op de veranderende informatiebehoefte van de organisatie. De kans dat een legacy-systeem opgebouwd is volgens het drie-lagenmodel is gering. Presentatielaag, functielaag en data laag zijn vaak vergaand met elkaar verweven. Om de functies van een legacy-systeem beschikbaar te stellen aan de buitenwereld kan (een deel van) de functielaag worden gesimuleerd met behulp van zogenoemde adapters. In paragraaf 4.2 komt een en ander aan de orde. Als applicaties gegevens nodig hebben van of moeten leveren aan andere applicaties neemt de onderlinge afhankelijkheid van de applicaties toe. Interface-awareness betekent ook dat applicaties niet mogen "stuklopen" omdat een andere applicatie tijdelijk niet beschikbaar is.

Domeinen

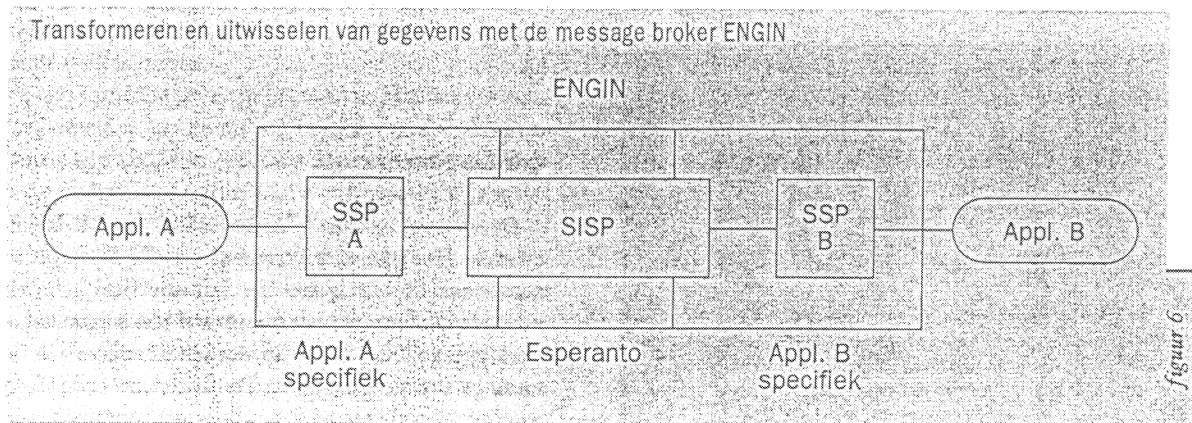
Als het aantal applicaties dat via de message broker wil communiceren erg groot wordt of als er bijvoorbeeld meerdere opdrachtgevers (met wellicht tegengestelde belangen) zijn, dan is het vanuit het oogpunt van beheer verstandig om meerdere (kopieën van) message brokers te installeren. Elke message broker krijgt dan zijn eigen domein waartoe een aantal applicaties behoort. Applicaties die sterk van elkaar afhankelijk zijn, moeten bij voorkeur tot hetzelfde domein behoren. Daardoor kan de communicatie tussen domeinen beperkt blijven. In paragraaf 5.2 wordt aangegeven dat de message broker ENGIN van de BU BC als domein gaat functioneren binnen het grotere geheel van KPN Telecom.

Single point of failure

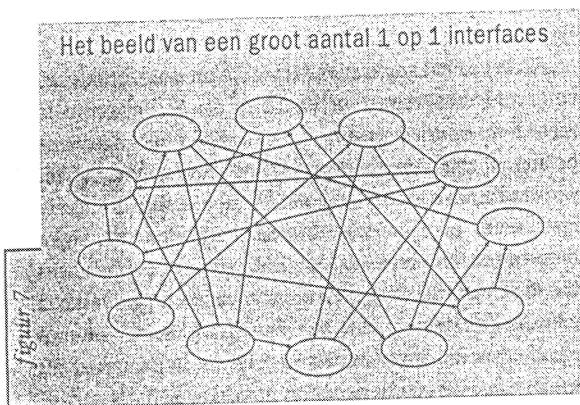
Het gebruik van een message broker voor het uitwisselen en transformeren van gegevens tussen applicaties is, ook uit oogpunt van beheer, een zeer verstandige zaak. Maar zoals zo vaak, kleven er aan de invoering van een message broker ook nadelen. Doordat de gegevensuitwisseling van veel applicaties via de message broker loopt, is er als het ware een "single point of failure" in de infrastructuur ontstaan. Als de message broker onderuit gaat, dan ligt alle communicatie tussen de applicaties stil. De risico's van een single point of failure kunnen tegenwoordig worden ingeperkt door het nemen van adequate (tegen)maatregelen. In dit artikel wordt hierop niet verder ingegaan.

Esperanto

Zoals in de probleemstelling is aangegeven heeft de BU BC teveel 1 op 1 interfaces (zie figuur 7) en brengt het onderhoud te hoge kosten met zich mee. Door interfaces



figuur 6

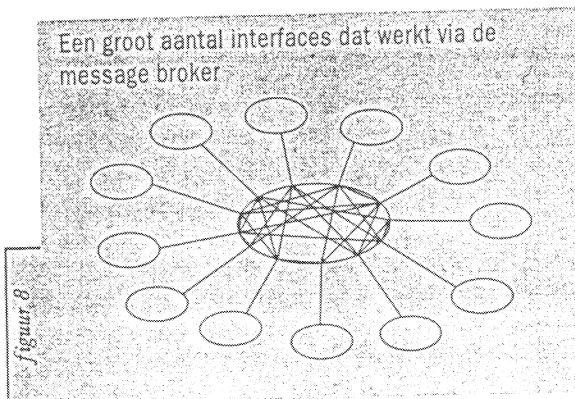


aan te sluiten op een message broker blijken de kosten sterk teruggebracht te kunnen worden, onder andere omdat:

- de interfaces kunnen worden gestandaardiseerd;
- hergebruik van oplossingen mogelijk is;
- kennis van het gebruik van de message broker gebundeld is, dit in tegenstelling tot de grote verzameling 1 op 1 interfaces waarvan doorgaans de bouwers niet meer beschikbaar zijn en het beheer niet altijd adequaat is ingericht;
- de message broker al veel werk kan verrichten tijdens het transformeren;
- beheer centraal kan plaatsvinden.

Maar hoe zit het met het aantal interfaces?

In principe verandert het aantal interfaces niet (zie figuur 8). Van iedere interface moet een SISP (transformatieroutine) gemaakt worden. Dit is in het algemeen een stuk eenvoudiger dan het vervaardigen van een 1 op 1 interface. Toch wil de BU BC naar minder interfaces toe. Daartoe heeft het Software Huis binnen de message bro-



ker een soort "Esperanto" (tussentaal) gedefinieerd. Alle data wordt eerst vertaald naar de Esperanto-vorm en vervolgens naar het gewenste "dialect" van de bestemmingsapplicatie. Dit betekent dat binnen de message broker een uniform datamodel wordt gehanteerd. Alle data worden daarbij tweemaal getransformeerd. Het aantal transformatieroutines kan daardoor worden beperkt tot $2 \cdot n$ (voor iedere applicatie zijn transformatieroutines van en naar het uniform datamodel nodig), als n het aantal aangesloten applicaties is. Dat is aanzienlijk minder (zie figuur 9) dan in het geval van directe transformaties waarin, net als bij 1 op 1 koppelingen, het maximum aantal transformatieroutines $n \cdot (n-1)$ bedraagt. Ook het onderhoud is minder arbeidsintensief. Indien van applicatie A, die met m andere applicaties een koppeling heeft, het interface-record wijzigt, moeten bij directe transformatie $2 \cdot m$ transformatieroutines worden aangepast terwijl in het geval van het gebruik van Esperanto slechts 2 transformatieroutines behoeven te worden aangepast. Alleen in het geval dat de wijzigingen dusdanig zijn dat het Esperanto, het uniforme datamodel, voor de betrokken applicaties moet worden aangepast moeten $2 \cdot m$ transformatieroutines worden aangepast. Dit laatste zal, indien het uniforme datamodel zorgvuldig is opgesteld, slechts in zeer beperkte mate gebeuren. In tabel 2 is een overzicht van een en ander gegeven.

Transport

Om berichten te kunnen versturen, moeten ze worden beschreven of populair gezegd "ingepakt". De ontvanger van een bericht kan het weer "uitpakken" als hij weet hoe de verpakking eruitziet. De huidige versie van ENGIN werkt met segmenten als beschrijvingsmethode. Ook ander standaarden (bijvoorbeeld EDI en HL7) werken op basis van segmenten. Ondersteuning door ENGIN van XML (eXtensible Markup Language) is sinds kort mogelijk.

4.2 DE PRAKTIJK

De message broker is sinds 1998 bij de BU BC in gebruik. Hoe zijn de ervaringen? De BU BC heeft in nauwe samenwerking met het Software Huis gekozen voor een kwalitatieve benadering en is zeer zorgvuldig te werk gegaan. Stap voor stap worden interfaces aan de message broker gekoppeld. De opgedane ervaringen worden gebruikt om tot een proces van continue kwali-

Tabel 2: Benodigd aantal transformatieroutines bij n applicaties

Benodigd aantal transformatieroutines bij n applicaties			
	directe transformatie	gebruik van een Esperanto	conventionele architectuur (1 op 1 kopp.)
max. aantal routines	$n*(n-1)$	$2*n$	$n*(n-1)$
min. aantal routines te wijzigen	$2*m$	2	$2*m$
max. aantal routines te wijzigen	$2*m$	$2*m$	$2*m$

n = het aantal applicaties
m = het aantal betrokken applicaties bij wijziging van een interface-record

teitsverbetering te komen. In de literatuur staat dit bekend als de cirkel van Deming ("Plan-Do-Check-Act"-cirkel) [Dorr1999]. Voor het aansluiten van de eerste interfaces is ruimschoots de tijd genomen. De snelheid van aansluiten neemt echter drastisch toe. Het voorbereiden, de functionele fase, kost doorgaans vrij veel tijd. Indien de functionaliteit van een interface goed gedefinieerd is, dan is de technische realisatie binnen de message broker doorgaans binnen enkele dagen of weken geregeld. Verwacht wordt dat het aantal koppelingen in het jaar 2001 sterk toeneemt.

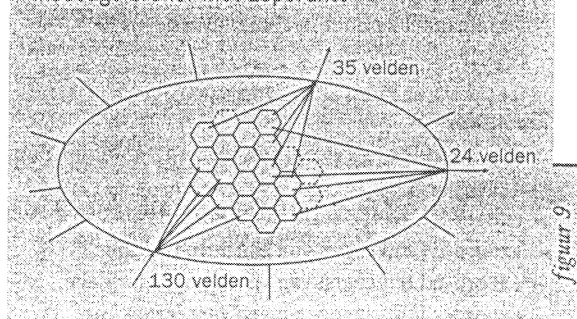
Om inzicht te geven in het soort applicaties dat thans gekoppeld is aan de message broker wordt hierna een aantal applicaties genoemd dat is aangesloten op de message broker. In paragraaf 4.2.1 wordt beschreven hoe een interface van M&M007 met FIT functioneert. In paragraaf 4.2.2 wordt beschreven hoe onlangs een interface tussen M&M007 en Baan ERP is gerealiseerd. Voorbeelden van applicaties die interfacen via de message broker zijn:

- *M&M007*: Meld- en Meetpost 007

Registratie en afhandeling van storingsmeldingen betreffende telecommunicatieapparatuur en -installaties;

- *FIT*: Field Information Terminal
Met behulp van FIT worden monteurs elektronisch aangestuurd. Daartoe beschikken de monteurs over een mobiele werkplek en een GSM toestel (de zogeheten "FIT-koffer");
- *CKR*: Centrale Klanten Registratie
CKR ondersteunt het klantgericht werken door een eenduidige identificatie van klanten mogelijk te maken en inzicht te bieden in de relatie tussen een klant en KPN Telecom;
- *SIZM*: Service Informatie Zakelijke Markt
SIZM bevat de prijzen van service-artikelen. Deze prijzen worden gebruikt in het servicewerkveld voor het doorberekenen van verleende services en verbruikte materialen;
- *CAPS*: Capaciteits Planning Systeem
Met behulp van CAPS wordt de beschikbare monteurscapaciteit beschikbaar gesteld aan de verkoop-eenheden;
- *KIOSK*: Klant Interactie Ondersteunend Systeem
KIOSK ondersteunt de eenheid "Sales Bedrijven" bij de verkoop, levering en facturering van standaardorders. In KIOSK worden gegevens betreffende orders, assortiment en klanten vastgelegd;
- *LVV*: Leveringssysteem Vaste Verbindingen
In LVV worden gegevens over alle verbindingen, klant- en werkorders in één bestand geregistreerd. De gegevens staan ter beschikking van de diverse districten/organisaties;
- *SVV*: Storingen Vaste Verbindingen
SVV is een trouble ticketing systeem voor klantklachten over vaste verbindingen;
- *NUMBES*: Nummer Beheer Systeem
NUMBES bevat de nummerdatabase van geporteerde

Een voorbeeld van een generieke interface via de message broker met Esperanto



- telefoonnummers met daarbij de operator waarnaar het telefoonnummer is geporteerd;
- AWO: Automatisering Werk Orders
- AWO ondersteunt het "consumentenproces" bij de verkoop, levering en facturering van standaardorders.

4.2.1 Koppeling van serviceorders en monteurs

Zoals in hoofdstuk 3 is vermeld, is M&M007 het informatiesysteem dat wordt gebruikt voor de registratie en afhandeling van storingsmeldingen betreffende telecommunicatieapparatuur en -installaties. M&M007 biedt ondersteuning bij het vertalen van een storing naar een serviceorder. M&M007 kan worden gebruikt om serviceorders af te handelen, maar het is ook mogelijk om serviceorders (door) te leveren aan een ander informatiesysteem, dat de betreffende serviceorders afhandelt en soms ook weer terugstuurt. Het FIT informatiesysteem ontvangt van M&M007 serviceorders ter afhandeling, door en ter ondersteuning van de monteurs, en ook Baan ERP ontvangt van M&M007 serviceorders. Om een en ander te regelen heeft M&M007 een "serviceorder interface" die een aantal transacties (zie tabel 1) omvat. De service/dienst die M&M007 met deze interface verleent, wordt beschreven met behulp van een Service SPecificatie (SSP). Ook de service die FIT verleent, wordt in een SSP beschreven. Het transformeren gebeurt door ENGIN op basis van de Service Implementatie SPecificatie (SISP). In het kort wordt nu ingegaan op de SSP en SISP.

Inhoud van het SSP "M&M007 serviceorder interface"

De volgende onderwerpen maken deel uit van het SSP:

- de betreffende transacties;
- validaties door de message broker ENGIN;
- validaties door M&M007;
- acties;
- foutafhandeling door het externe systeem;
- berichtenspecificatie.

Transacties

Transacties om serviceorders te laten verwerken door externe systemen:

- ext-ord-1: Leveren van een serviceorder;
- ext-ord-2: Ontvangen van een terugmelding;
- ext-ord-3: Ontvangen van voortgangsinformatie;
- ext-ord-4: Ontvangen van een gereedmelding.

Transacties om serviceorders (intern) te verwerken door M&M007:

- int-ord-1: Ontvangen van een serviceorder;
- int-ord-2: Leveren van voortgangsinformatie;
- int-ord-3: Leveren van een gereedmelding;
- int-ord-4: Leveren van informatie over de serviceorder.

Met behulp van de transactie ext-ord-1 meldt M&M007 een serviceorder aan bij een extern informatiesysteem, bijvoorbeeld FIT. Het betreffende externe informatiesysteem blijft eigenaar van de serviceorder totdat het een terug- of gereedmelding aan M&M007 stuurt. Het externe informatiesysteem kan gedurende de afhandeling van de serviceorder rapporteren over de voortgang.

Validaties

Door de message broker ENGIN wordt een beperkt aantal validaties uitgevoerd. De belangrijkste validatie is: "Het bericht moet voldoen aan het in ENGIN gespecificeerde formaat". M&M007 stelt duidelijke eisen aan het "Ontvangen van een serviceorder" (transactie int-ord-1). Tabel 3 geeft enkele validaties weer.

Acties/Foutafhandeling

Als alle validaties tot een goed resultaat leiden voert M&M007 een aantal activiteiten uit om de serviceorder op te slaan. Het externe informatiesysteem kan in het veld "resultaatcode" zien of de serviceorder succesvol is opgeslagen of dat er fouten zijn opgetreden.

Berichtenspecificatie

In de berichtenspecificatie wordt aangegeven hoe het interface-record van de "M&M007 serviceorder interface" eruitziet.

Inhoud van het SISP "M&M007 - FIT serviceorder interface"

In de SISP worden de mappings, waarmee records van M&M007 worden afgebeeld op records van FIT vastgelegd. Tabel 4 bevat enkele velden van de mapping van de transactie "Leveren van een serviceorder" door M&M007 aan FIT. Het werkelijk aantal velden bedraagt 130 (zie figuur 9).

Tabel 3: Enkele validaties van de transactie "Ontvangen van een serviceorder"

Validatie	Foutcode
Het externe informatiesysteem moet in M&M007 geautoriseerd zijn voor deze transactie	100
De in het bericht als verplicht gemarkeerde velden moeten gevuld zijn	200
Als "contractsoort" gevuld is, dan moet "berekend tijdstip" ook ingevuld zijn en vice versa	206
De "code productsoort" moet bestaan in het gegevensblok "assortiment"	400

Tabel 4: Enkele velden van de mapping van de transactie "Leveren van een serviceorder" door M&M007 aan FIT

Veld van M&M007	Veld van FIT
MELD_S.klantorder nummer	KLA_ID
MELD_S.telefoonnummer	KLA_TCNR
MELD_S.product-ID	KLA_LIJNBENA
VERPL_S.urgentie code	URG_CODE
WERK_S.uitgifte datum	KLA_DT_UTTG
KLANT_S.klantnaam	KLA_NAAM
WERK_S.analist code	ANA_CODE
MELD_S.order type	KLA_INFO

DE MENING VAN ... XANDRA KOOL

Xandra Kool was tot voor kort werkzaam bij het "Coördinatiecentrum services" van de Productline Ericsson (PLE) van de BU BC. Zij was als keyuser intensief betrokken bij het pilotproject PERFECTO dat tot doel had het "vererpen" van de informatievoorziening binnen de PLE.

Een essentieel onderdeel van PERFECTO was het realiseren van koppelingen van Baan ERP met bestaande informatiesystemen van PLE. Om een tussentijdse beoordeling van het project mogelijk te maken wenste de stuurgroep op een bepaald moment een demonstratie, inclusief de interfacemogelijkheden.

De functionele specificaties van de interfaces waren op dat moment reeds opgesteld. Binnen twee weken werkten de interfaces met de informatiesystemen M&M007 (call registratie) en FIT (opdrachten voor de Field Engineers), die met Baan ERP het proces "Uitvoeren correctief onderhoud n.a.v. een call" ondersteunen.

Xandra: "De snelheid waarmee een en ander is gerealiseerd heeft mij aangenaam verrast. Dit was mede

mogelijk omdat de applicaties M&M007 en FIT reeds door middel van een generieke interface met de message broker waren verbonden. Daardoor behoeften deze applicaties niet te worden gewijzigd. Wijzigen van deze applicaties zou, in verband met de, zeer zorgvuldig werkende, change procedure een aanzienlijk langere doorlooptijd hebben betekend. Naar mijn mening is dit een duidelijke verbetering van de slagkracht van de informatievoorziening.

Toch heb ik nog wel wat op te merken. Twee andere applicaties waren via een 1-op-1 verbinding met elkaar gekoppeld. Soms ging er iets fout. Aangezien ik de applicaties en de technische omgeving door en door kende, had ik wat trucjes achter de hand om de productie toch doorgang te kunnen laten vinden.

Sinds enige tijd zijn de betreffende applicaties gekoppeld via de message broker. Een enkele keer gaat er iets fout met de verbinding. In die situaties moet ik nu contact opnemen met "interfacebeheer" die het probleem verhelpt. Vanuit KPN bekeken is het natuurlijk beter dat het "interfacebeheer" is geregeld en dat de technische kennis van de interfaces geconcentreerd is in een team rondom de message broker. Maar ik vind het wel jammer van mijn trukendoos!"

4.2.2 Koppeling van serviceorders met Baan ERP

Ook Baan ERP ontvangt serviceorders van M&M007. Daartoe is een SSP voor Baan ERP gemaakt en een SISP voor de mapping van het M&M007 serviceorder-record op het betreffende Baan ERP-record. Een groot voordeel was dat aan de kant van M&M007 geen wijzigingen nodig waren. Wijzigingen aanbrengen in een dergelijke cruciale applicatie vergt doorgaans een aanzienlijke doorlooptijd, die door de gekozen aanpak, een *generieke interface* met SSP, kan worden voorkomen. Daardoor was het mogelijk de gewenste koppeling in korte tijd te realiseren. De benodigde tijd ging vooral zitten in het aansluiten van de nieuwe applicatie Baan ERP.

ERP pakketten staan niet bekend om hun openheid met betrekking tot het communiceren met andere applicaties. Baan Company heeft met versie V, ook wel bekend onder de naam Correlli, een substantiële stap in de richting van openheid gedaan [Baan1999]. Met versie V is het mogelijk om *on line* te communiceren. Daartoe is er in versie V sprake van een lagenstructuur. Het is nu mogelijk om ook op een andere manier dan via on line Baan-sessies met Baan ERP te communiceren. Daartoe heeft men zogenaemde BOI's gedefinieerd. Een Business Object Interface maakt het mogelijk om op een andere

manier dan via de on line Baan-sessies, met behoud van de Baan-functionaliteit, te communiceren met Baan ERP. De snelheid van het realiseren van een koppeling hangt sterk af van de beschikbaarheid van de noodzakelijke BOI. Als deze reeds gebouwd is, kan de koppeling snel worden gerealiseerd. In het andere geval zal de BOI door de leverancier gebouwd moeten worden, hetgeen een zekere doorlooptijd met zich meebrengt. Naast de BOI's is er sprake van DAL's (Dynamic Access Layer). Een DAL fungeert als communicator met de Baan-database (-tabellen). Door deze mogelijkheden kan een externe applicatie communiceren met Baan ERP zonder kennis te hebben van de tabellenstructuur van Baan ERP. Als de ontwikkelafdeling wijzigingen in de tabellenstructuur aanbrengt hoeft dat voor de BOI's en externe applicaties geen gevolgen te hebben.

Adapter

In paragraaf 4.1 is aangegeven dat als een applicatie (een deel) van zijn functielaag beschikbaar wil stellen, dat kan worden gesimuleerd met behulp van adapters. Figuur 6 kan daartoe worden uitgebreid met een adapter. In figuur 10 is dit weergegeven.

Om de ENGIN-Baan adapter te kunnen laten werken zijn ingrediënten (waaronder C-componenten en Java-

DE MENING VAN ... BENNY LANKHORST

Benny Lankhorst is werkzaam bij het "Internet Center" van het Software Huis van KPN Telecom. Hij heeft in 1999 de opdracht gekregen een web-interface te bouwen voor het informatiesysteem M&M007 (call registratie). Op die manier kunnen "calls" via Internet worden aangeboden aan M&M007.

Benny: "Eindgebruikers zijn in feite niet geïnteresseerd hoe en waarmee een applicatie wordt gebouwd. Wel zijn ze geïnteresseerd in de snelheid waarmee een applicatie beschikbaar komt en de kwaliteit van de applicatie. Alvorens in te gaan op de realisatie kan ik stellen dat het gebruik van de message broker ervoor heeft gezorgd dat de betreffende eindgebruikers de web-interface sneller beschikbaar kregen dan was gepland en eerder dan verwacht. Bovendien hebben de gebruikers opvallend weinig last van storingen. Zij zijn dan ook zeer tevreden.

Als bouwers hadden we een aantal voordelen. We konden ons volledig richten op de web-technologie en hoefden ons niet te bekommeren om de legacy-applicatie die ongewijzigd kon blijven. De moeilijkheids-

graad van de applicatie is daardoor beperkt gebleven. Als twee applicaties interfacen via een 1-op-1 koppeling dan ontstaat er bij een storing vaak een vervelende situatie omdat beide applicatieleveranciers de schuld bij de ander leggen. Nu de applicaties via de message broker interfacen is er als het ware een neutrale component waarmee snel kan worden vastgesteld welke applicatie de storing veroorzaakt. Ik ben ervan overtuigd dat interfacing met behulp van een message broker niet alleen aanzienlijk voordeliger is dan via traditionele koppelingen, maar dat bovendien de beheerkosten sterk afnemen.

Ik ben voorstander van kleinschalig beginnen en dan stap voor stap uitbouwen in plaats van een werkwijze waarbij een concept dat zich nog niet bewezen heeft in één klap grootschalig wordt geïntroduceerd. Ik voel mij dan ook happy met de gekozen aanpak, met de message broker, door het Software Huis en de BU BC. Wat ik niet begrijp is dat er zo weinig over dit onderwerp gepubliceerd wordt. Wie weet er nou dat het allemaal zo handig is?"

componenten) van zowel KPN en ENGIN als Baan ERP gebruikt. Figuur 11 geeft dit weer en laat ook de positie van de DAL's zien.

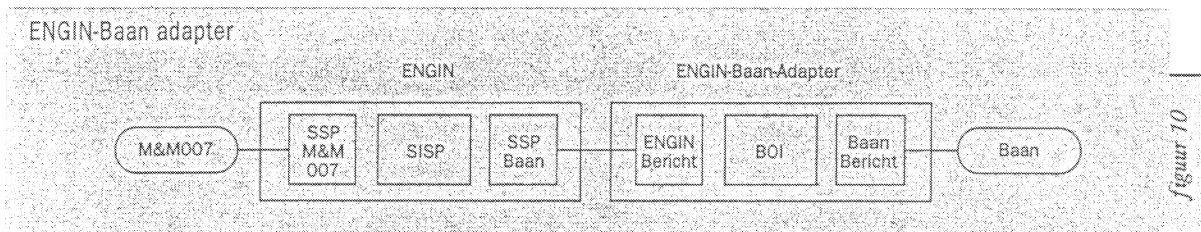
5 Toekomstige ontwikkelingen

5.1 XML

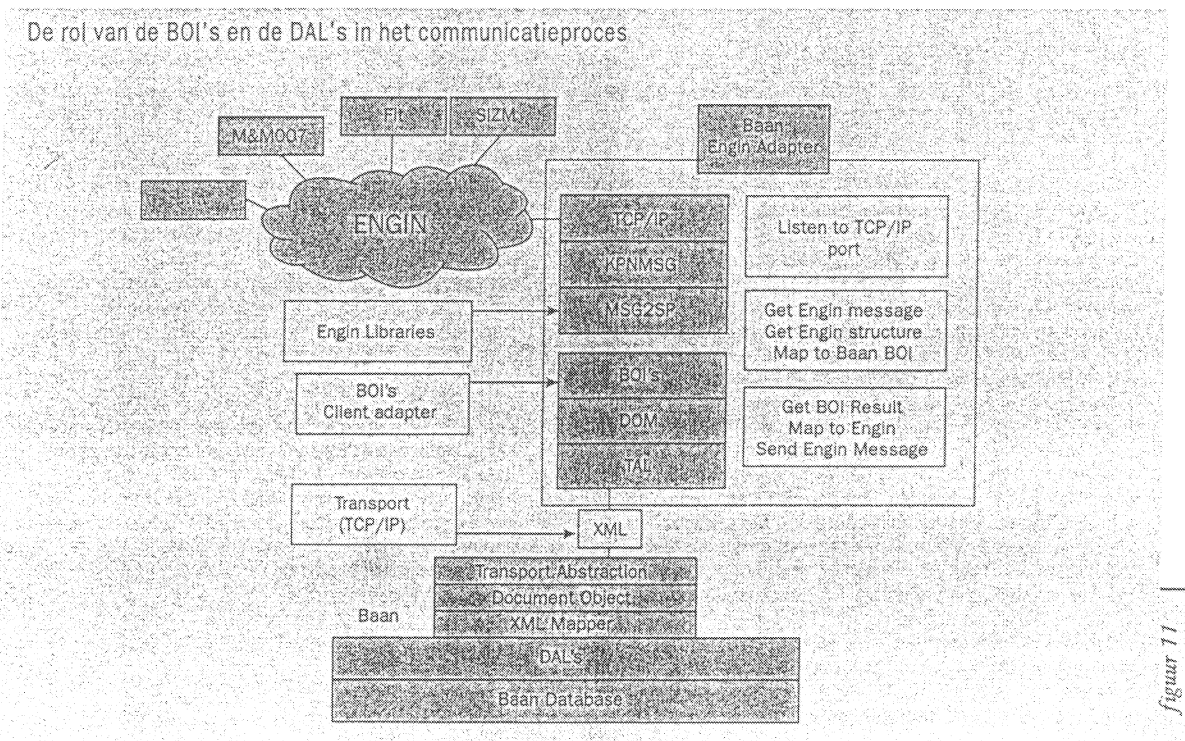
XML, de eXtensible Markup Language is een eenvoudige en zeer flexibele taal die gebaseerd is op SGML (Standard Generalized Markup Language). XML maakt het voor informatieconsumenten en -producenten veel eenvoudiger met elkaar te communiceren. Gegevensuitwisseling tussen applicaties die op verschil-

lende platformen draaien of gebruikmaken van verschillende besturingssystemen kan worden gerealiseerd met XML. XML biedt uitgebreide middelen voor het beschrijven van de inhoud van documenten en van metagegevens. Verwacht wordt dat XML binnen de web-technologie HTML gaat vervangen. De praktijk zal moeten uitwijzen of XML echt een standaard wordt of dat toch weer conversie van de ene XML-omgeving naar de andere XML-omgeving nodig is.

Binnen Baan ERP wordt reeds gewerkt met XML en ook de nieuwe versie van ENGIN ondersteunt XML. Verwacht wordt dat binnenkort binnen BU BC de communicatie via de message broker met behulp van XML plaatsvindt.



figuur 10



figuur 11

5.2 GAIA

In dit artikel is ingegaan op de casus van de BU BC waarbij de gekozen oplossingen en ervaringen aan de orde zijn gekomen. Parallel aan deze activiteiten wordt binnen KPN Telecom gewerkt aan het project GAIA. GAIA staat voor "Gedistribueerde Applicatie Infrastructuur Architectuur". Dit project wordt op corporate level uitgevoerd. Zodra GAIA operationeel is, sluit de BU BC zich aan bij GAIA en vormt met haar message broker ENGIN een domein binnen GAIA (zie figuur 12). Voor GAIA is het voordeel dat bij de start direct een omvangrijk en goed functionerend domein beschikbaar is.

5.3 MAATSCHAPPELIJK RELEVANTE INTERFACING

In dit artikel staat het onderwerp interfacing van applicaties centraal. Het gaat daarbij voornamelijk om de interfacing van applicaties binnen een grote organisatie. Een stap verder is het interfacen van applicaties over organi-

saties heen. In zijn column in IT-Monitor 1/2000 signaleert prof. Nielen [Niel2000] dat na 50 jaar automatisering op basis van een "technology push" heel voorzichtig de "demand pull" boven de horizon komt. "Vragen vanuit de overheid en bedrijfsleven naar oplossingen voor bestaande problemen. Een daarvan betreft het volgende. Informatiesystemen maken gebruik van datadefinities. Wanneer onafhankelijk ontwikkelde informatiesystemen – later – met elkaar moeten communiceren, moet de ene datadefinitie geprojecteerd worden op de andere. Een voorbeeld is het verlangen van de overheid om een aanvraag van sociale bijstand te toetsen aan gegevens van bedrijfsverenigingen en van de belastingdienst. Dergelijke communicatie is vooralsnog razend moeilijk". Nielen is van mening dat een universele interface-generator ontwikkeld moet worden. De auteurs van dit artikel zijn van mening dat de in dit artikel beschreven oplossingen en ervaringen een substantiële bijdrage kunnen leveren om tot een dergelijke interface-generator te komen.

DE MENING VAN ... MARC GOFFERJÉ

Marc Gofferjé is manager van de Business Proces Group van de Business Unit BedrijfsCommunicatie.

Marc: "De Business Unit BedrijfsCommunicatie (BU BC) van KPN Telecom opereert zelfstandig als een winstgevendende entiteit op de telefoniemarkt. De Business Proces Group (BPG) heeft als kerntaak: het inrichten van, door ICT ondersteunde, processen ten behoeve van de BU. De BU werkt met smalle marges en moet daarom 'op de centen passen'. Daarom is het uit financiële overwegingen niet altijd haalbaar om alle KPN-ICT-standaarden te gebruiken en is de BU voortdurend op zoek naar betere, snellere en goedkopere ICT-oplossingen.

Wij willen modulair automatiseren en voeren 'ontkoppelen' hoog in het vaandel. Ontkoppelen heeft een aantal facetten. Eén van die facetten is het afkoppelen van bepaalde KPN-brede legacy-systemen die 'te groot en te duur' zijn voor onze BU. Andere systemen moeten worden aangekoppeld. Bij nieuwe systemen hanteren we als richtsnoer:

- kopen; - schaalbaarheid; - zo min mogelijk aanpassingen; - koppelbaarheid (minimale interface-complexiteit).

Een derde facet van 'ontkoppelen' is het voorkomen

dat dezelfde functionaliteit in meerdere systemen voorkomt ('ontdubbelen'). Om een en ander te realiseren maken we sinds 2 jaar gebruik van de message broker ENGIN. Dat bevalt ons zeer goed. De interface-complexiteit is significant afgenomen. Nieuwe interfaces met reeds gekoppelde systemen kunnen veel sneller worden gerealiseerd. Zo bedroeg de doorlooptijd van het realiseren van een interface met ons systeem M&M007 voorheen ongeveer 7 maanden. Tegenwoordig is een en ander binnen 4 weken operationeel.

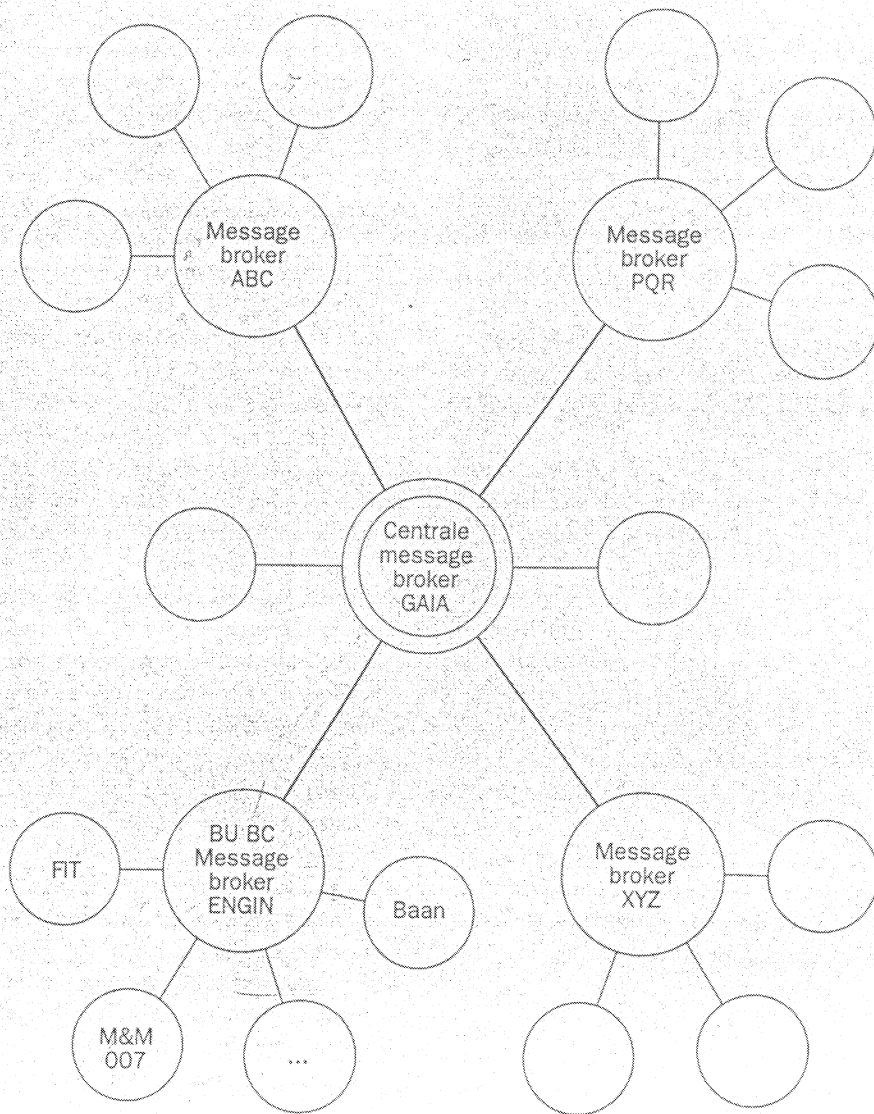
Door de nieuwe werkwijze kunnen wij niet alleen veel sneller reageren op verzoeken uit de gebruikersorganisatie, maar zijn ook de omvangrijke interfacekosten duidelijk afgenomen. Belangrijk is ook dat lijnmanagers die een nieuwe interface willen laten vervaardigen expliciet moeten aangeven welke kostenbesparingen met deze interface worden bereikt. Wat de message broker betreft: wij zetten de gekozen koers met kracht voort!".

6 Conclusie

In hoofdstuk 2 is als probleemstelling geformuleerd: "Er zijn teveel 1 op 1 interfaces en het onderhoud van de interfaces brengt te hoge kosten met zich mee". Het probleem kon als volgt worden aangepakt:

- 1 realiseer zoveel mogelijk nieuwe interfaces via de message broker;
- 2 vervang bij groot onderhoud van de huidige interfaces deze door interfaces via de message broker, tenzij er door het management goed te keuren redenen zijn om dat niet te doen;
- 3 rapporteer over de voor- en nadelen van het gebruik van de message broker.

De message broker ENGIN van BU BC als domein binnen GAIA



figuur 12

De BU BC is sinds 1998 bezig met het realiseren van nieuwe interfaces via de message broker (punt 1). Bij groot onderhoud van bestaande interfaces worden deze vervangen door interfaces via de message broker, waarbij de betreffende interface generiek wordt gemaakt (punt 2). Dit artikel draagt bij aan punt 3.

De conclusie van de BU BC is dat het gebruik van de message broker, voorzien van een Esperanto-faciliteit, het aantal 1 op 1 koppelingen substantieel vermindert, de kosten van realisatie en onderhoud sterk doet afnemen en een adequaat beheer van de interfaces mogelijk maakt.

De Business Unit BedrijfsCommunicatie zet met betrekking tot de gegevensuitwisseling/interfacing van applicaties met kracht de ingezette koers voort.

De conclusie wordt toegelicht aan de hand van een aantal voordelen, nadelen en richtlijnen.

6.1 VOORDELEN

- een message broker is een applicatie die volledig ingericht is voor interface management waardoor het eenvoudiger, en goedkoper, is om interfaces op elkaar af te stemmen;
- door het toepassen van een message broker kan elke applicatie in zijn eigen formaat met de buitenwereld communiceren. Hierdoor wordt het eenvoudiger om applicaties aan te passen of te vervangen;
- als iedere applicatie nog maar één interface met de buitenwereld hoeft te hebben, betekent dit dat de architectuur van de gehele "enterprise" gaat veranderen. Het maximum aantal interfaces neemt af van $n*(n-1)$ naar $2*n$;
- bundeling van kennis op één plaats;
- onafhankelijk (centraal) beheer van de interfaces wordt mogelijk (en is gewenst);
- het is mogelijk om een grote mate van standaardisatie met betrekking tot het transformeren in te voeren;
- koppelingen op data laag-niveau, die de data-integriteit in gevaar brengt, worden voorkomen.

6.2 NADELEN

- als de message broker onderuit gaat, ligt alle communicatie tussen de applicaties stil (single point of failure);
- toename van de onderlinge afhankelijkheid van applicaties;
- de neiging bestaat om functionaliteit, die in applicaties thuishoort, te verplaatsen naar de message broker.

6.3 RICHTLIJNEN

- om het interface management goed te stroomlijnen is het nodig een beheerproces in te richten dat applicatieonafhankelijk is. Het beheerproces moet een leidende rol spelen bij het definiëren en ondersteunen van koppelingen en moet voldoende bevoegdheden hebben om een leidende rol in het interface management in te vullen. Daarnaast moet er commitment van alle betrokken partijen zijn om koppelingen via de message broker te laten lopen;
- de applicatie die "eigenaar" van de gegevens is, bepaalt de recordlayout van het bericht;
- de message broker kan een vraag opsplitsen in meerdere vragen indien het antwoord uit verschillende systemen moet komen. Vervolgens kunnen de antwoorden weer samengevoegd worden tot één antwoord. Men moet bij het gebruik van deze functionaliteit wel oppassen dat men geen business functionaliteit in de message broker neerlegt;
- als twee applicaties gegevens uitwisselen, is het belangrijk te weten wat het doel van de communicatie is en welke rol de betrokken applicaties hebben. Dit bepaalt welke acties uitgevoerd worden en wat de inhoud van het antwoordbericht kan zijn. Zonder deze informatie kan een message broker zijn rol als "tussenstation" niet correct vervullen;
- de applicaties moeten bij voorkeur opgebouwd zijn volgens de drie-lagen-architectuur;
- de applicaties moeten er rekening mee houden dat sommige gegevens uit andere applicaties tijdelijk niet beschikbaar zijn;
- door adapters te bouwen kan, ten behoeve van communicatie, een functielaag worden gesimuleerd;
- koppelingen moeten uitgevoerd worden op het niveau van de functielaag. Data laag-koppelingen zijn niet toegestaan;
- applicaties die sterk van elkaar afhankelijk zijn, moeten

tot hetzelfde domein behoren;

- uit oogpunt van continuïteit is het noodzakelijk om adequate maatregelen te nemen om de consequenties van een single point of failure te beperken;
- voor sommige bestaande maatwerkkoppelingen die veel business functionaliteit bevatten, kan het zinvol zijn deze niet via een message broker te koppelen.

Literatuurlijst

- [Baan1999] Baan Development, *Baan EAI - Open World, Positioning White paper*, Baan Company, December 1999, 20 pag.
- [Clau1999] Claus, M. *Inventarisatie, Functies en gegevens landelijke systemen BU BC*, KPN Telecom, BU BC, mei 1999, 37 pag. (intern rapport)
- [Dec1998] Dec, K., R. Mack and E. Andren, *Transition Strategies, From Mainframes to Distributed Computing*, The Technical Issues, Gartner Group, February 1998, 76 pag.
- [Dorr1999] Dorr, D. en J. Zuidema, *Werken met het model Nederlandse Kwaliteit, De Quick Scan van uw organisatie*, Kluwer Bedrijfsinformatie, 1999, 87 pag.
- [Dijk1984] Dijk, A.J. van, R.P. Karlas, P. Makkes en R.J.J. van Schie, 'GEMS'- *GEautomatiseerd Magazijnadministratie- en voorraadbeheersysteem, Functioneel ontwerp*, Technische Universiteit Delft, Memorandum no RC-ISO-84 004, juni 1984, 256 pag.
- [Dijk2000] Dijk, A.J. van, R.E. Overeem, Th. Stolker en J. Weber, *Interface Baan - ENGIN, Technisch Ontwerp*, KPN Telecom / Baan Company, april 2000, 24 pag. (intern rapport)
- [Ecks1999] Eckstein, Robert, *XML*, Pocket Reference, O'Reilly, Cambridge, oktober 1999, 107 pag.
- [GALA2000] Projectgroep GALA, *Basisonderwerp GALA*, KPN Telecom, januari 2000, 47 pag. (intern rapport)
- [Musc1999] Muscato, *ENGIN v4.0, Message Broker & WebEnabler*,

Muscato Corporation, Altamonte Springs, 35 pag.

[Niel1989] Nielen, G.C., *Omgaan met georganiseerde informatiesystemen*, Handboek Informatica, A1500, Samsom Bedrijfsinformatie, 1989, 24 pag.

[Niel2000] Nielen, G.C., *Innovatie in de wereld*, IT-Monitor, nr. 1, 2000, 1 pag.

[Otze2000] Otzen, E., *Message broker ENGIN, M&M007 service order interface*, KPN Telecom, februari 2000, 58 pag. (intern rapport)

[Over1998] Overeem, R., *Message broker, De inzet van een message broker*, KPN Telecom, juli 1998, 38 pag. (intern rapport)

[Pit1999] Pitts, Natanya, *XML*, Sybex, Soest, juni 1999, 543 pag.

[Rock2000] Rock-Evans, R., *Kunnen IDL en EAI samen?*, Software Release Magazine, februari 2000, pag. 11-12.

[Stee1997] Steenbergen, M.E. van, *Mapping, Een handvat voor bestissingen*, KPN Telecom, BU BC, mei 1997, 17 pag. (intern rapport)

Ir. A.J. van Dijk RE RI is werkzaam als onafhankelijk IT-adviseur/IT-auditor bij Avédé-Info b.v. te Zoetermeer. Hij is bij de Business Unit BedrijfsCommunicatie van KPN Telecom betrokken bij architectuurvraagstukken en bij de invoering van Baan ERP.

Ing. R.E. Overeem is werkzaam als senior software engineer bij Logica b.v. te Groningen. Hij is bij het Software Huis van KPN Telecom betrokken bij de realisering van message broker toepassingen.

J.J. Schijf is werkzaam als hoofd ontwikkeling bij de Business Proces Group van de Business Unit BedrijfsCommunicatie van KPN Telecom. Hij heeft in 1998 het initiatief genomen om de message broker ENGIN aan te schaffen.

auteur

Het ontwikkelen en bouwen van een verzameling informatiesystemen is te vergelijken met het varen tussen Scylla en Charybdis. Als er te veel aan het eigen initiatief van bedrijfsonderdelen wordt overgelaten dan verzandt een organisatie in eiland-automatisering met hier en daar 1-op-1 verbindingen tussen de systemen. Als echter gestreefd wordt naar een geïntegreerd en monolithisch informatiesysteem, dan loopt men op de klippen van de complexiteit.

De auteur van dit casus-artikel reikt ons, op basis van praktijkervaringen, een slimme tussenweg aan: de "message broker". De "message broker" is een informatiesysteem dat als een makelaar andere informatiesystemen met elkaar verbindt. Van Dijk gaat echter nog een stap verder met de introductie van een zogenaamde Esperanto-faciliteit die de uitkomsten van systemen kan vertalen in een vorm die voor elk ander systeem begrijpelijk is. Het belang van deze aanpak voor organisaties die moeten werken met een mengvorm van veel oude en nieuwe informatiesystemen is groot. Interessant is ook de conclusie dat de verantwoordelijkheid voor de ontwikkeling en het beheer van de "message broker" een andere is dan die voor de bouw en ontwikkeling van andere informatiesystemen.

gegevensuitwisseling

interfaces

message broker

Baan ERP

**SOFTWARE
RELEASE**

www.release.nl

Magazine

JAARGANG 6 • JUNI 2001 • NUMMER 4 • f 18,50

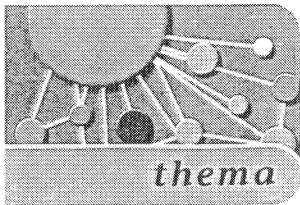
VAKBLAD OVER SOFTWARE-ONTWIKKELING

Enterprise Application *Integration*

EAI meer dan click, drag en drop
**Goede communicatie-infrastructuur
voorwaarde voor EAI**
EAI en ERP met Baan Open World
Microsoft talks business

Voorbespreking van Delphi 6

**Performant programmeren in Java (2): tips voor snellere code - .Net in Lego-
land: Microsoft's nieuwe marketecture - Rock-Evans: XA en de DTP-service**



Dit artikel gaat over de realisatie van adapters tussen BaanERP en de message broker Engin. Eerst wordt een korte inleiding over de casus bij KPN Telecom gegeven en wordt een kader voor EAI geschetst. Daarna komt de realisatie van de Baan-Engin-Adapter aan de orde.

EAI en ERP met Baan Open World

De realisatie van een adapter in de praktijk van de Business Unit BedrijfsCommunicatie van KPN Telecom

KPN Telecom is in Nederland een zeer belangrijke speler op de telecommunicatiemarkt en moet snel en flexibel inspelen op de dynamiek in de telecommunicatiemarkt en de steeds veranderende wensen van de klanten. KPN doet dat door een gedifferentieerd producten- en dienstenaanbod. Daartoe is de organisatie van KPN ingedeeld in onder andere Business Units, Verkoop-eenheden en Productie Eenheden. Eén van de Business Units is BedrijfsCommunicatie (BU BC). De BU BC houdt zich bezig met de inkoop, verkoop, installatie en service van alle apparatuur en systemen die bij klanten worden geplaatst. In het werkveld van de BU BC zijn enkele duizenden mensen werkzaam: veel monteurs, planners en werkvoorbereiders maar ook marketeers, financiële medewerkers en personeelsfunctionarissen. Het doel van de BU BC is marktleider zijn en blijven en het realiseren van een goed rendement. Om dat doel te bereiken werkt de BU continu aan het verbeteren van de bedrijfsprocessen en de daarbij ondersteunende ICT. De organisatiestructuur van KPN laat toe dat Business Units een zelfstandig IT-beleid voeren, mits er gebruik gemaakt wordt van corporate data, zoals klantgegevens.

Sinds augustus 2000 wordt door de business line BedrijfsCommunicatieSystemen (BCS) van de BU BC gewerkt aan het inrichten en implementeren van BaanERP software. Dat gebeurt binnen het project Mecano (Met ERP-Concept Aangestuurde Nieuwe Organisatie).

- BaanERP zal niet alle bestaande informatiesystemen vervangen. Daarom is het noodzakelijk dat BaanERP op adequate wijze communiceert met een aantal van de overige bedrijfsinformatiesystemen. Binnen Mecano is het Kolibrie-team (KPN On Line Interfacing met BaanERP.

Resultaat Is Effectief) verantwoordelijk voor de realisatie van deze communicatie door middel van interfaces. In dit artikel staat de interface van BaanERP met het Centrale KlantRegistratiesysteem (CKR) en de wijze waarop deze met behulp van een message broker en een adapter is gerealiseerd, centraal.

EAI Applicatie-integratie komt in verschillende vormen voor. Deze diversiteit maakt dat een standaard integratieoplossing niet bestaat. Enkele vormen van applicatie-integratie zijn [Elsw2001]:

- gegevens uit verschillende gegevensbronnen moeten worden gecombineerd om nieuwe gegevens af te kunnen leiden. Applicatie-integratie krijgt dan de vorm van gegevensintegratie. Dat is het gebied van de data-warehouses;
- applicaties moeten samenwerken om nieuw resultaat te kunnen bereiken. Daarbij wordt de functionaliteit van verschillende systemen geïntegreerd;
- applicaties vormen onderdeel van een keten. Ieder heeft haar eigen functionaliteit, maar de resultaten of gemeenschappelijk benodigde gegevens moeten worden doorgegeven aan de volgende applicatie in het bedrijfsproces. In deze context wordt vaak de term enterprise application integration (EAI) gebruikt. De keten kan ook over de grenzen van een organisatie heen gaan, bijvoorbeeld richting andere bedrijven (Business to Business).

In dit artikel komt applicatie-integratie in de laatste vorm (EAI) aan de orde. Applicatie-integratie vereist dat de uiteenlopende representaties uit verschillende bronnen op elkaar kunnen worden afgebeeld. Dit vereist een soort gemeenschappelijke taal waarin vorm en betekenis van

de gemeenschappelijke concepten worden vastgelegd: semantische integratie. Een actuele syntactische (vorm) taal is XML (eXtensible Markup Language).

Vanuit de technologische invalshoek is EAI ook het proces waarmee, via een netwerk dat de machines van een of meer bedrijven met elkaar verbindt, zowel applicaties als gegevensbestanden kunnen worden geïntegreerd. De applicaties en de gegevensbestanden kunnen gebouwd zijn met een diversiteit aan technologieën. Daarbij vereist de integratie weinig tot geen wijzigingen van de bestaande applicaties en/of gegevens, oftewel een 'inbreukvrije' benadering [Rock2000]. EAI ontleent zijn enorme belang aan het feit dat het een vorm van op componenten gebaseerde ontwikkeling mogelijk belooft te maken. De belangrijkste componenten van een EAI-architectuur zijn (zie figuur 1):

- **middleware:** de middleware levert de onderliggende ondersteuning voor gedistribueerde verwerking en zorgt door het gehele netwerk heen voor de veilige en betrouwbare aflevering van berichten en aanroepen;
- **adapters en connectoren:** adapters zorgen voor de wederzijdse vertaling tussen de verschillende technologische niveaus, zoals het vertalen van de aanroepen berichtopmaak van de intern gerichte opmaak naar de opmaak zoals die door een door derden geproduceerd product wordt ondersteund en vice versa. Connectoren worden gebruikt om transactieverwerkingsprotocollen te vertalen;
- **message brokers:** Gartner [Dec1998] geeft als definitie en doel van de message broker: "Message brokers are logical hubs that copy and forward messages to one or more destinations. A message broker is an intelligent third party (hence a 'broker') between information sources and information consumers. The purpose of this architectural type is to integrate independently designed application domains (e.g., new applications, purchased applica-

tions and legacy applications). Message brokers make it possible to re-engineer business processes without re-engineering all the application programs and databases";

- **workflow:** de workflowsoftware bestuurt het proces. Dit houdt bijvoorbeeld in dat een gebruiker bepaalde informatie invoert, waarna de workflow zorgt voor het activeren van de message broker, applicaties en adapters.

INTERFACES Een mogelijke definitie van een interface is: een koppeling tussen (twee) applicaties met als doel het uitwisselen van gegevens. Het ontwikkelen van een interface kan worden gezien als het ontwikkelen van een beperkt informatiesysteem. Daarbij worden de volgende stappen onderkend:

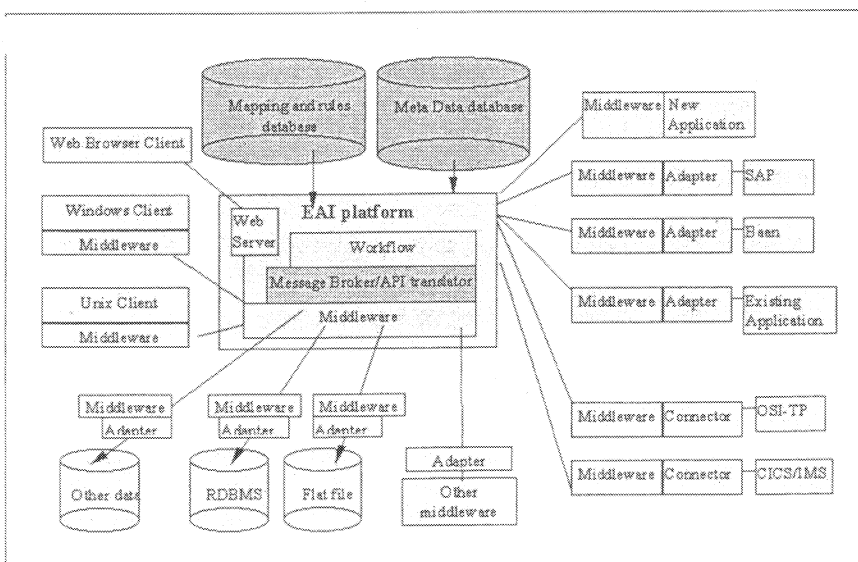
- indienen van een verzoek om een interface te realiseren;
- houden van een intake-gesprek;
- uitvoeren een impactanalyse;
- opstellen van een Functioneel Interface Document;
- opstellen van een Technisch Ontwerp;
- realiseren;
- testen;
- installeren;
- beheren.

Een complicerende factor bij interfaces is dat er altijd meerdere partijen bij betrokken zijn.

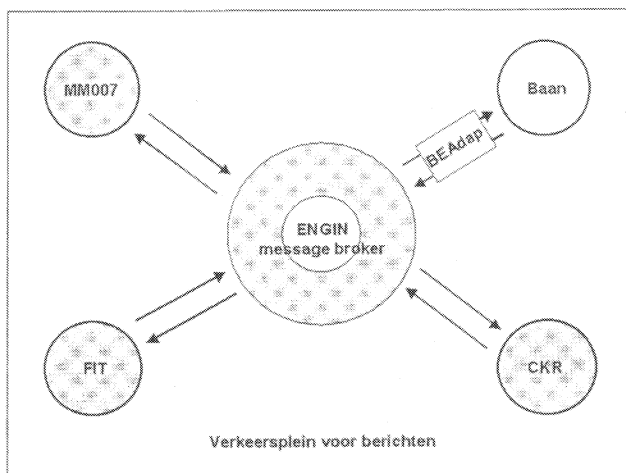
INVULLING EAI BIJ HET PROJECT MECANO Binnen de BU BC en het project Mecano wordt EAI onder andere ingevuld met de message broker Engin van de firma Muscato/Optio Software. Om BaanERP te koppelen met de overige bedrijfsinformatiesystemen wordt gebruik gemaakt van Baan Open World en van adapters.

De BU BC heeft de message broker Engin drie jaar geleden aangeschaft. Belangrijke motieven daarvoor waren enerzijds het beter beheersbaar te maken van het beheer van interfaces en anderzijds het sterk omlaag brengen van de kosten.

Met de komst van de message broker is een proces ingezet, waarbij nieuwe interfaces zoveel mogelijk via de message broker worden gerealiseerd en bestaande interfaces bij groot onderhoud worden vervangen door interfaces via de message broker. Op die manier moet het aantal 1 op 1 interfaces, die hoge kosten met zich meebrengen, sterk worden teruggebracht. Over de ervaringen met Engin is zeer recent een uitvoerig artikel gepubliceerd [Dijk2001]. Naast de con-



FIGUUR 1: Componenten van een EAI-architectuur (bron: Rock-Evans)



FIGUUR 2: Message broker Engin en de Baan-Engin-Adapter binnen de EAI van de BU BC

clusies over het gebruik zijn hierin voordelen, nadelen en richtlijnen gegeven.

Enkele richtlijnen zijn:

- het is niet gewenst om business functionaliteit in de message broker te beleggen;
- applicaties moeten bij voorkeur zijn opgebouwd volgens de drie-lagen-architectuur;
- koppelingen van applicaties moeten worden uitgevoerd op het niveau van de functielaag;
- door adapters te bouwen kan, ten behoeve van communicatie, een functielaag worden gesimuleerd.

Figuur 2 geeft een (beperkte) weergave van de EAI-architectuur van de BU BC. De message broker Engin fungeert als "verkeersplein voor berichten". BaanERP communiceert via een Baan-Engin-Adapter (kortweg BEAdap) en Engin met andere applicaties zoals CKR.

Een derde component binnen de EAI-architectuur is Baan Open World (kortweg BOW). BOW is een integratie-framework om Baan Enterprise Solutions applications, waaronder BaanERP, onderling te laten communiceren of te laten communiceren met andere applicaties. Binnen de EAI van BU BC wordt BOW gebruikt om BaanERP te laten communiceren, via Engin en BEAdap's, met de overige bedrijfs-informatiesystemen. BOW bestaat onder andere uit de volgende componenten (zie figuren 3 en 5):

- een aantal BOW-adapters, waaronder een BaanERP Open World Adapter;
- Business Object Interfaces (BOI's). Een BOI is een standaardmethode (API) om Baan Open World objecten via de BOW-adapters te verbinden met andere applicaties. Een BOI maakt het mogelijk om op een andere manier dan via de online BaanERP-sessies, met behoud van de BaanERP-functionaliteit, te communiceren met BaanERP. Binnen BaanERP bestaat een Business Object uit een of meer gegevenstabellen en programma's (DAL, DLL) om de gegevens te bewerken. BaanERP heeft een aantal standaard BOI's. Indien nodig kunnen aanvullende BOI's worden gemaakt;

- Baan Open World Studio, de ontwikkelomgeving om BOI's te creëren;
 - XML, dat als hulpmiddel wordt gebruikt om koppelingen tussen applicaties te vereenvoudigen of stroomlijnen;
 - Baan Open World Services, waaronder de transport-service (op O/S-laag en netwerklaag) waarmee het berichtenverkeer tussen twee applicaties wordt geregeld.
- Baan Open World is aanvullend op middleware en is ontworpen om te communiceren met een groot aantal transportprotocollen en producten.

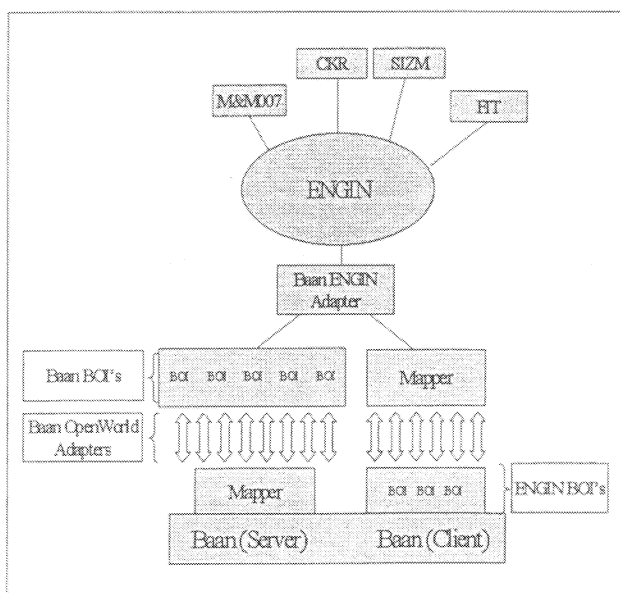
DE BAAN-ENGIN-ADAPTER VOOR CKR Functioneel Ontwerp van de interface BaanERP - CKR

CKR is een corporate informatiesysteem waarmee binnen KPN Telecom klantgegevens worden geregistreerd. Andere informatiesystemen kunnen gebruik maken van deze centraal beheerde klantgegevens. Daartoe heeft CKR een aantal transacties beschikbaar zoals:

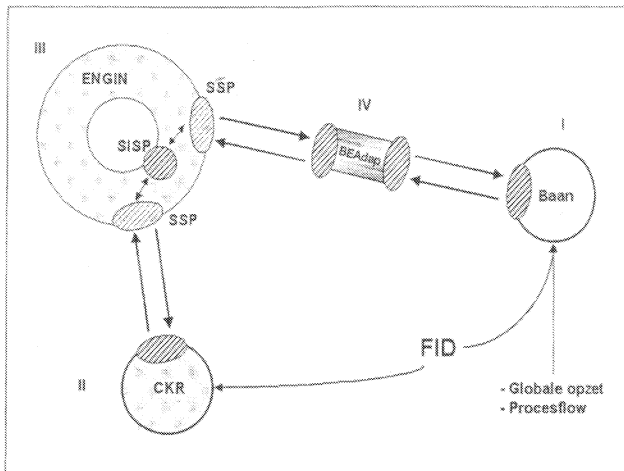
- klantgegevens opvragen uit CKR;
- het aanbrengen van een verwijzing bij een klant. Daarmee wordt een abonnement op mutaties genomen. Dagelijks worden gegevensmutaties van de klanten waarop een abonnement is genomen via batchverwerking door CKR aan de abonnees aangeboden;
- het verwijderen van een verwijzing.

Indien gewenste klantgegevens nog niet in BaanERP zijn opgenomen, dan worden deze gegevens met behulp van een interface tussen BaanERP en CKR opgevraagd bij CKR en opgeslagen in BaanERP.

Hoe komt de interface tot stand? Na het uitvoeren van een impactanalyse wordt een Functioneel Interface Document (FID) opgesteld. Naast een korte inleiding van het betreffende proces komen gegevensoverdracht en selectiecriteria aan de orde. Het FID is het Functioneel Ontwerp van de interface. Het FID wijkt af van het Functio-



FIGUUR 3: Message broker Engin en Baan Open World



FIGUUR 4: De interface BaanERP - CKR

neel Ontwerp van een "gewoon" informatiesysteem doordat het bij een interface gaat om meer dan één informatiesysteem. In dit geval gaat het om een interface (gegevensuitwisseling) tussen CKR en BaanERP. In figuur 4 is de positie van het FID aangegeven. Het FID is bij deze interface uitgebreid met de documenten "Globale opzet" en "Procesflow". Deze documenten beschrijven hoe met de klantgegevens in BaanERP wordt omgegaan.

TECHNISCH ONTWERP VAN DE INTERFACE BAAN-ERP - CKR Het Technisch Ontwerp van een interface bestaat uit een aantal componenten (zie figuren 4 en 5):

- een technisch deelontwerp van de vragende applicatie, in dit geval BaanERP,
- een technisch deelontwerp van de leverende applicatie, in dit geval CKR,
- een technisch ontwerp van de oplossing binnen de message broker Engin,
- een technisch ontwerp van de Baan-Engin-Adapter/CKR.

Figuur 4 laat zien dat alle gearceerde delen een bijdrage leveren aan het Technisch Ontwerp.

BAANERP Zoals aangegeven is het FID van de interface BaanERP-CKR uitgebreid met een "Globale opzet". Hierin is met name beschreven hoe binnen BaanERP moet worden omgegaan met CKR/klantgegevens. Door de BU BC worden binnen BaanERP aanvullende klantgegevens, zoals een CKR-nummer opgeslagen. Sommige van deze gegevens worden als zoekargument gebruikt. Dit heeft geleid tot aanpassingen in BaanERP. Daartoe is een technisch deelontwerp gemaakt.

CKR CKR is gemaakt om diensten betreffende klantgegevens te verlenen. Daartoe is een aantal standaardtransacties gedefinieerd waarmee de diensten kunnen worden afgenomen. In feite kunnen deze transacties worden beschouwd als CKR-BOI's. De interface BaanERP-CKR

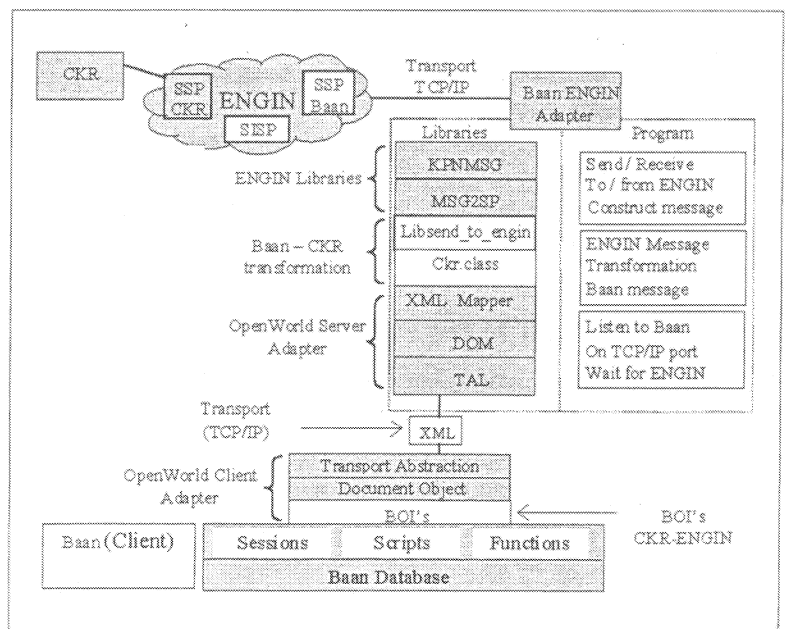
is gerealiseerd met behulp van een viertal CKR-BOI's. Aanpassingen aan CKR, bijvoorbeeld een extra transactie/BOI, waren niet nodig.

ENGIN De communicatie van applicaties met de message broker Engin verloopt via een Service SPecificatie (SSP). In een SSP wordt vastgelegd welke gegevens worden aangeboden en welke controles en transformaties moeten plaatsvinden. In feite kunnen SSP's worden gezien als Engin-BOI's.

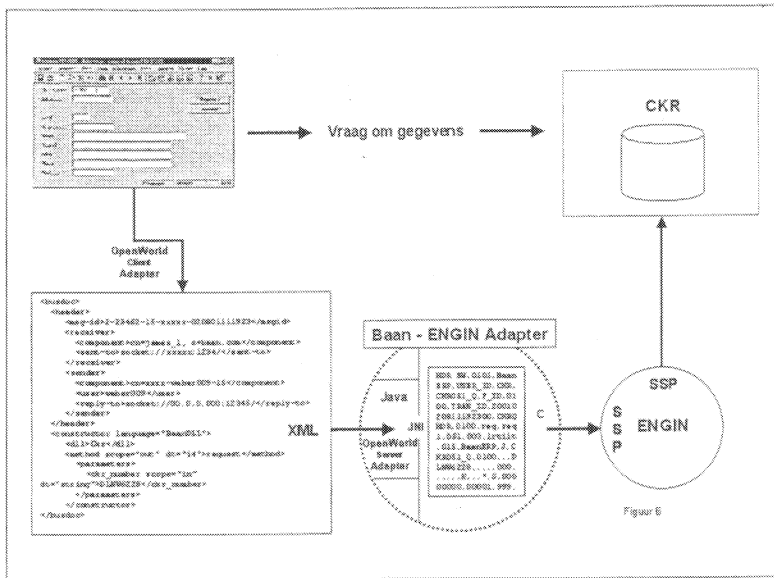
Engin zorgt voor het afbeelden van de gegevens van de ene applicatie op die van de andere applicatie. Daartoe wordt een Service Implementatie SPecificatie (SISP) opgesteld.

BEADAP/CKR De BEADAP/CKR zorgt voor een wederzijdse vertaling tussen Engin en BaanERP. Binnen de versie van Engin die bij de BU BC in gebruik is, wordt nog niet gewerkt met XML, maar met segmenten als beschrijvingsmethode. BOW is gebaseerd op XML. In de volgende paragraaf wordt in vogelvlucht de werking van de BEADAP/CKR beschreven.

DE WERKING VAN DE BEADAP/CKR Het realiseren van de BEADAP/CKR was een interessante activiteit omdat de twee omgevingen Engin en BaanERP nog niet eerder met elkaar hadden samengewerkt. Engin fungeert bij de BU BC al enkele jaren op basis van segmenten. ERP systemen zijn lange tijd gesloten geweest. Alle belangrijke bedrijfsgegevens werden in ERP gestopt. De laatste tijd bestaat bij veel bedrijven de behoefte om ERP-systemen te combineren met Best of Breed applicaties of eigen applicaties. BOW is ontwikkeld om te kunnen voldoen aan de door klanten noodzakelijk geachte communicatie van BaanERP met andere informatiesystemen. KPN Telecom



FIGUUR 5: Technisch ontwerp BaanERP - CKR



FIGUUR 6: Flow verzoek van BaanERP aan CKR

is één van de eerste bedrijven die gebruik maakt van BOW. Na uitvoerige discussies is besloten om de adapter niet onder te brengen bij Engin of BaanERP, maar als een afzonderlijk proces te realiseren. Belangrijke kwaliteitsaspecten daarbij waren flexibiliteit en aanpasbaarheid. Verwacht mag worden dat nieuwe ontwikkelingen, zoals XML, nieuwe (vereenvoudigde) communicatiemogelijkheden bieden. Het vervangen van een afzonderlijk proces is dan eenvoudiger dan een proces dat sterk verweven is met de message broker of met BaanERP. De BEAdap bestaat uit de volgende componenten (zie figuur 5):

- een mechanisme om naar Engin te luisteren (het "Engin-oor");
- een mechanisme om naar BaanERP te luisteren (het "BaanERP-oor");
- een deel (deel 1) dat tot doel heeft berichten van BaanERP te ontvangen of aan BaanERP aan te bieden;
- een deel (deel 3) dat tot doel heeft berichten van Engin te ontvangen of aan Engin aan te bieden;
- een deel (deel 2) dat de communicatie (gegevens transformeren en doorgeven) tussen deel 1 en deel 3 verzorgt.

Op een aantal punten wordt nu kort ingegaan.

XML De bij de BU BC werkende versie van Engin werkt nog niet met XML. Om de gegevensstructuur van de Engin-BOI (de SSP) bekend te maken binnen de BEAdap en binnen BaanERP wordt een opgeleverde SSP omschreven naar een XML-bericht. Dit bericht wordt door een XML-mapper vertaald naar een Java-omgeving ten behoeve van deel 1 van de BEAdap en naar een

Baan 3 GL-omgeving ten behoeve van BaanERP. Op die manier is de gegevensstructuur bekend binnen alle componenten.

DEEL 3 - ENGIN Het Engin-deel bevat de Engin Libraries. De functionaliteit in deze libraries is ontwikkeld door het Software Huis van KPN en is geschreven in de programmeertaal C.

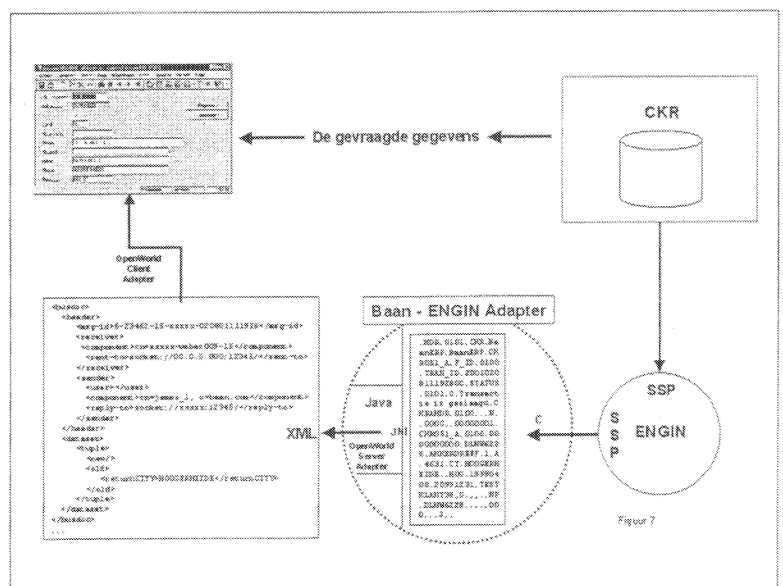
DEEL 1 - BAANERP Het BaanERP-deel bevat (een deel van) de BOW-adapters. De gehanteerde programmeertaal is Java. De BOW-adapters communiceren op basis van XML met BOW-adapters aan de BaanERP kant. Afhankelijk van de initiatiefnemer vervullen ze de rol van BOW-server-adapter of BOW-client-adapter.

DEEL 2 De communicatie tussen deel 1 (Java) en deel 3 (C) gebeurt met behulp van Java Native Interface (JNI).

ENGIN-OOR Het Engin-oor is een deelproces dat luistert op een poort naar berichten die verwerkt moeten worden. Deze functionaliteit wordt beschikbaar gesteld door de Engin Libraries.

BAANERP-OOR Het BaanERP-oor is een deelproces dat luistert op een poort naar XML-berichten vanuit BOW. Deze functionaliteit maakt deel uit van de BOW-adapters.

DE FLOW VAN HET OPVRAGEN EN VERWERKEN VAN EEN CKR-BERICHT Figuur 6 laat zien hoe een verzoek vanuit een BaanERP-sessie aan CKR via de BEAdap/CKR en Engin verloopt. Het CKR-nummer van de klant wordt door de betreffende sessie van BaanERP aangeboden aan BOW. De BOW-client-adapter biedt een XML-



FIGUUR 7: Flow gegevens van CKR naar BaanERP

Afkortingen/acroniemen

API	Application Programming Interface
BEAdap	Baan-Engin-Adapter
BO	Business Object
BOI	Business Object Interface
BOW	Baan Open World
BU BC	Business Unit BedrijfsCommunicatie
CKR	Centrale KlantenRegistratie
DAL	Data Access Layer
DCD	Document Contents Description
DLL	Dynamic Link Library
DOM	Document Object Module
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
FID	Functioneel Interface Document
ICT	Informatie- en CommunicatieTechnologie
JNI	Java Native Interface
Kolibrie	Kpn On Line Interfacing met BaanERP. Resultaat Is Effectief
Mecano	Met Erp-Concept Aangestuurde Nieuwe Organisatie
QCD	Query Contents Definition
SISP	Service Implementatie SPecificatie
SSP	Service SPecificatie
XML	eXtensible Markup Language
TAL	Transport Abstraction Layer

bericht aan op een poort. De BOW-server-adapter, die onderdeel uitmaakt van de BEAdap/CKR, luistert naar deze poort en pikt het bericht op. Met behulp van een Java-programma wordt, via JNI, het CKR-nummer doorgegeven aan een C-programma dat de gegevens aanbiedt aan een (Engin-)poort. De message broker pikt de gegevens op en stuurt deze naar de applicatie CKR.

Figuur 7 toont hoe de gegevens vanuit CKR via Engin en de BEAdap/CKR op het scherm van de BaanERP-gebruiker komen. CKR zet de gegevens die behoren bij het betreffende CKR-nummer klaar voor Engin. De message broker pikt de gegevens op en geeft ze, via de (Engin-)poort door aan de BEAdap/CKR. Het C-programma stuurt de gegevens, via JNI, door naar de BOW-server-adapter. Deze biedt de gegevens door middel van XML-berichten aan op de poort. De BOW-client-adapter pikt de XML-berichten op en via een sessie van BaanERP komen de CKR-gegevens op het scherm van de BaanERP-gebruiker. De gebruiker kan ze, na een visuele controle, met behulp van de BaanERP-sessie in de database opslaan.

CONCLUSIE Het neerzetten van een infrastructuur waarmee de communicatie tussen BaanERP en CKR adequaat verloopt, via Baan Open World en een BEAdap en de message broker Engin, heeft een substantiële inspanning vereist. Met name het op elkaar laten aansluiten

van de diverse componenten heeft de nodige inspanning gevraagd en zal verder worden geoptimaliseerd. Nieuwe BEAdap's kunnen tamelijk snel worden gerealiseerd omdat de BEAdap/CKR hergebruikt kan worden. Alleen het datadeel moet worden aangepast. Figuur 5 geeft aan welke onderdelen dat zijn (de witte onderdelen). Inmiddels zijn enkele BEAdap's gerealiseerd. Het komende jaar zal nog een aantal BEAdap's in gebruik worden genomen. De verwachting bestaat dat nieuwe versies van Engin, gebaseerd op XML, en BOW de communicatie zal vereenvoudigen.

Ir. Aart J. van Dijk RE

is werkzaam als onafhankelijk IT-adviseur/IT-auditor bij Avédé-Info b.v. te Zoetermeer. Hij is bij de Business Unit BedrijfsCommunicatie van KPN Telecom betrokken bij architectuurvraagstukken en bij de invoering van BaanERP.

Drs. Jeroen Weber

was tot voor kort werkzaam als Senior International Technical Consultant bij Baan Nederland N.V. en was bij KPN Telecom betrokken bij de invoering van BaanERP. Hij is thans werkzaam bij SeeBeyond te Hoofddorp.

Literatuur

- [Baan1999] Baan Development, Baan EAI – Open World, Positioning White paper, Baan Company, December 1999
- [Dec1998] Dec, K., R. Mack and E. Andren, Transition Strategies, From Mainframes to Distributed Computing, The Technical Issues, Gartner Group, February 1998
- [Dijk2001] Dijk, A.J. van, R.E. Overeem en J.J. Schijff, Gegevensuitwisseling (interfacing) van applicaties met behulp van een message broker, De applicaties van de Business Unit BedrijfsCommunicatie van KPN Telecom, IT Management [Select], jaargang 7, nummer 1, mei 2001, pp. 39-57
- [Elsv2000] Elswijk, M. van en M. Maat, Aspecten van een integratiestrategie, Informatie, november 2000, pp. 44-48
- [Musc1999] Muscato/Optio Software, ENGIN v4.0, Message Broker & WebEnabler, Muscato Corporation, Altamonte Springs, 1999
- [Rock2000] Rock-Evans, R., Kunnen IDL en EAI samen?, Software Release Magazine, februari 2000, pag. 11-13.
- [Wage2001] Wagemans, R. en M. Baan, Functioneel Interface Document BaanERP → CKR, Project Mecano, deelproject Kolibrie, KPN Telecom, februari 2001 (intern rapport)
- [Webe2001] Weber, J. and A.J. van Dijk, Technical Design Interface Baan - CKR, Project Mecano, subproject Kolibrie, KPN Telecom, March 2001 (intern rapport)