6.2.3.D Wat weten wij van onze klanten?

Drs. ing. Mark van der Pas en drs. Rien Hamers

Samenvatting

Het is niet onze bedoeling in dit artikel het potentiële belang van CRM voor uw bedrijf uit de doeken te doen. Het meemaken van de nodige hypes heeft ons duidelijk gemaakt dat IT'ers niet de aangewezen personen zijn een businesscase hard te maken. Dat is een taak voor de afdelingen Marketing en Verkoop. Op die afdelingen weten ze het best hoe de relatie met de klant te optimaliseren. Maar stel nou dat Marketing met het verzoek komt de informatievoorziening zó aan te passen dat het bedrijf de relatie met klanten kan managen. Eén van de eisen van Marketing is in dat geval de beschikking over een integraal klantbeeld. De uitdaging is dan een aantal onderdelen uit de informatiehuishouding in het algemeen en gegevensmanagement in het bijzonder te professionaliseren. Dit artikel gaat over die laatste uitdaging.

Eén integraal klantbeeld

Een *integraal* beeld van een klant ontstaat wanneer een organisatie alle beschikbare gegevens van een persoon of bedrijf in zijn rol als (potentiële) klant samenvoegt. Aan welke gegevens moet je dan denken? Natuurlijk aan de naam- en adresgegevens, aan de afgenomen producten en diensten en aan de lopende bestellingen. Maar ook aan de reacties van de klant op de geleverde goederen, zoals zijn betalingsgedrag, de mate waarin hij de beschikbare functies van het product gebruikt, het onderhoud dat hij erop laat plegen en de vragen en klachten die hij heeft over het product. Die vragen en klachten komen niet alleen via de post en call centers binnen, maar ook via winkels en accountmanagers. De laatste zijn overigens een belangrijke bron van informatie over die klant. En steeds meer wordt die informatie (wanneer is wie bezocht, wat waren belangrijkste punten uit een gesprek et cetera) opgeslagen in een Sales Force Automation systeem. Andere groeiende bronnen van informatie over een klant zijn afkomstig uit het bezoek van de klant aan de internetsite van een organisatie en uit de e-mail die hij stuurt. Natuurlijk kan het zijn dat naast de beschikbare gegevens additionele klantgegevens nodig zijn, wellicht

INFORMATIE MANAGEMENT OKT. 2000

gegevens die niet of niet onmiddellijk beschikbaar zijn. Die gegevens zullen in dat geval apart vergaard moeten worden om te komen tot het benodigde integrale klantbeeld.

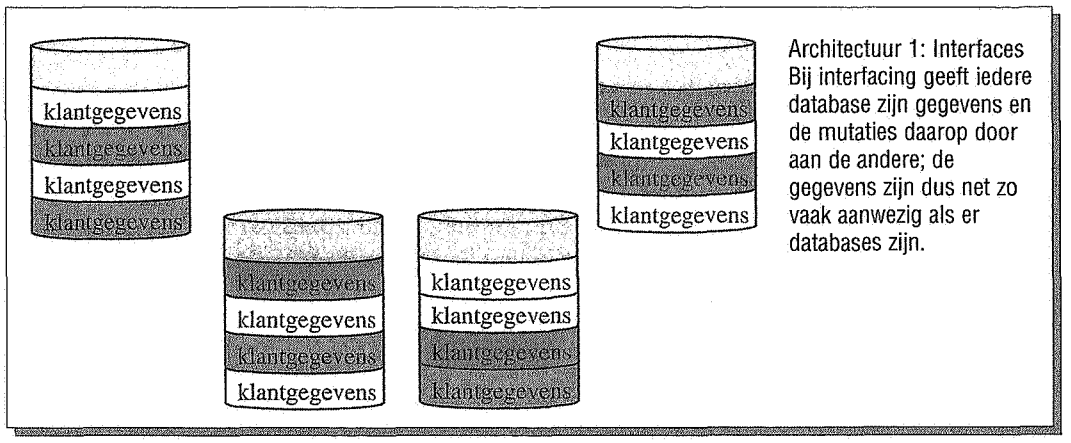
Vier IT-architecturen

De bottomline is dat één integraal beeld van een klant moet ontstaan, terwijl de IT-architectuur veelal zo is dat de daarvoor benodigde gegevens verspreid zijn over verschillende databases. Om de verspreide gegevens om te zetten in dat integrale klantbeeld kennen wij vier IT-architecturen. De kerneigenschappen van deze architecturen worden in de volgende paragrafen beschreven. Doordat 'de principes' van de architecturen zijn beschreven, komen deze in sommige gevallen weinig praktisch over. Wij hopen dat u ons dat vergeeft en het, na het lezen van dit artikel, met ons eens bent dat de essentie van de architecturen begrepen moet worden om ze succesvol te kunnen toepassen.

Figuur 1
Diverse applicaties

Architectuur 1: interfaces

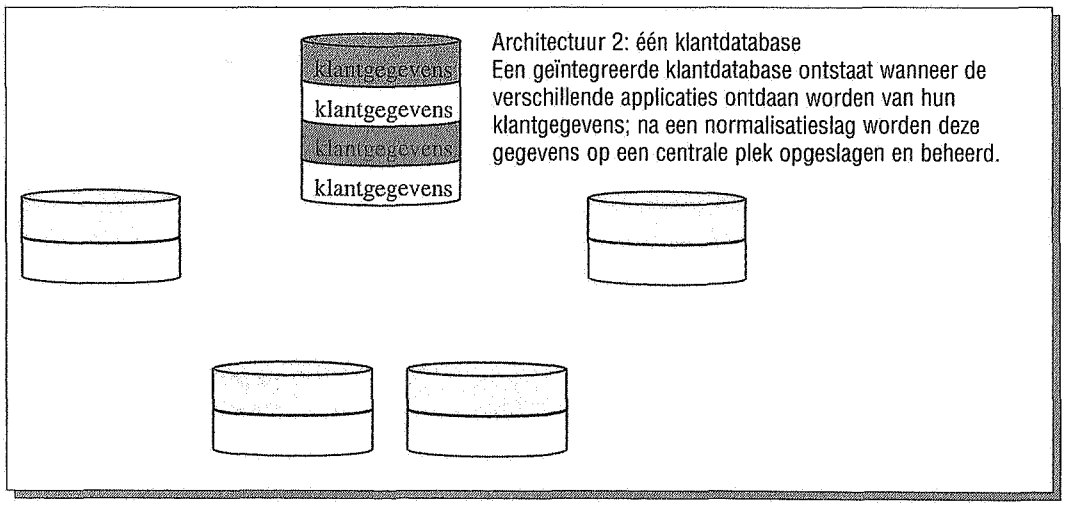
In de eerste architectuur worden in elk systeem die gegevens opgeslagen die de gebruikers van dat systeem nodig hebben. Om over de gewenste informatie te beschikken worden de gegevens tussen de verschillende systemen uitgewisseld. In zo'n omgeving worden, in het meest extreme geval, alle klantgegevens in elk systeem vastgelegd. In dat geval kan elk systeem een integraal klantbeeld bieden.
De standaardmanier om gegevens uit te wisselen is door interfaces te bouwen. Een geavanceerde manier om gegevens uit te wisselen tussen systemen maakt gebruik van *messaging*. In een messaging-omgeving is bekend waar bepaalde gegevens worden gebruikt en daarmee ook in welk systeem gegevens als adres- en telefoongegevens zijn opgeslagen. Wanneer in één van de systemen een adresgegeven wijzigt, stuurt het systeem een message naar de overige systemen die ook adresgegevens bevatten. Op basis van dat bericht wijzigen die andere systemen de adresgegevens in hun eigen database.



Figuur 2
Interfaces

Architectuur 2: één-klantdatabase

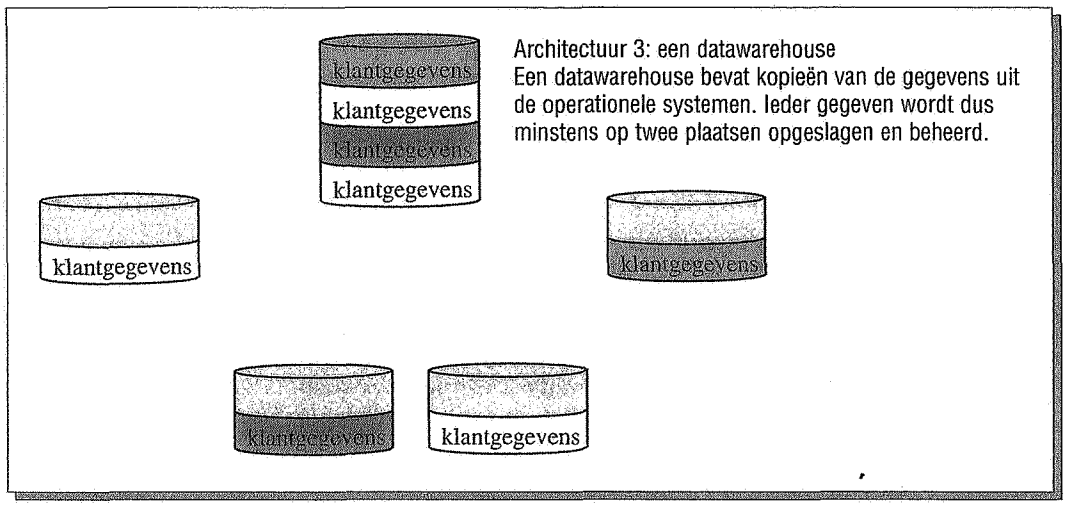
In principe bevat de applicatie-infrastructuur al het klantbeeld dat we willen construeren. Omdat de gegevens verspreid zijn over verschillende databases, beschikken we echter over gegevensverzameling waarin enkele modelleringsfoutjes zijn gemaakt. Immers, die omgeving is alles behalve genormaliseerd. Architectuur 2 ontstaat wanneer we zo'n normaliseringslag toepassen. Er ontstaat een nieuwe omgeving waarin klantgegevens nog maar op één plek worden opgeslagen en niet meer voor iedere applicatie afzonderlijk. De klantgegevens staan dan in eigen tabellen en alle applicaties verwijzen naar die centraal opgeslagen en bijgehouden klantgegevens.



Figuur 3
Eén-klantdatabase

Architectuur 3: datawarehouse

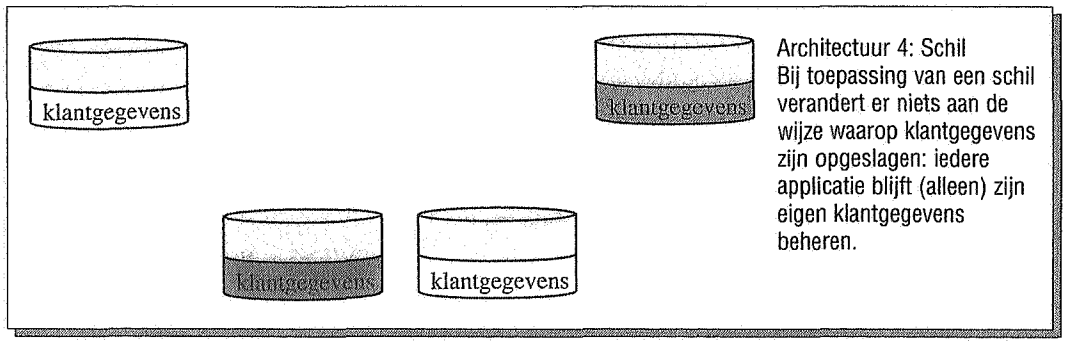
Een derde architectuur heeft als kenmerk dat gegevens uit de bestaande systemen worden gekopieerd in een nieuwe database. Een populaire naam voor die laatste database is een 'datawarehouse'. In deze architectuur worden klantgegevens minimaal twee keer opgeslagen. Eenmaal in het datawarehouse en minimaal eenmaal in onderliggende systemen.



Figuur 4
Een datawarehouse

Architectuur 4: schil

De laatste ons bekende architectuur kenmerkt zich niet door haar databasestructuur. Zij heeft als kenmerk dat het integrale klantbeeld gevormd wordt door de gegevens uit de verschillende databases te combineren op het moment dat daar bij een klantcontactmoment behoefte aan is. Daartoe wordt software gebruikt die de gegevens ophaalt uit de databases en samenvoegt tot het integrale



Figuur 5
Schil

beeld. Die software vormt een *schil* tussen de bestaande systemen en de gebruikers van de informatie. De databasestructuur is afhankelijk van de invulling van dit concept. In beginsel kan op alledrie de andere architecturen een schil worden gepositioneerd.

Vergelijking

Een vergelijking tussen de vier beschreven ideaaltypische architecturen brengt enkele duidelijke verschillen aan het licht. Die verschillen moeten een rol spelen wanneer u kiest welke architectuur u gaat inzetten.

Afweging 1: consistentie van klantgegevens

Een eerste belangrijk punt van overweging heeft betrekking op de consistentie van de gegevens. Zo kenmerkt een klantdatabase zich door het feit dat gegevens niet dubbel worden opgeslagen. Daarmee vermijdt zo'n database inconsistentie van klantgegevens. In de andere architecturen worden klantgegevens meermalen opgeslagen. Daardoor kan inconsistentie optreden. Dat geldt ook voor een datawarehouse-architectuur. Het toepassen van een datawarehouse waarborgt immers niet dat de gegevens in de onderliggende systemen consistent zijn.

Afweging 2: geïntegreerde omgeving voor gebruikers

Wanneer we vanuit de stoel van de gebruiker naar de architecturen kijken, zien we nogal wat verschillen. Een schil heeft bijvoorbeeld als voordeel dat de gebruiker in één omgeving niet alleen een integraal klantbeeld kan verkrijgen, maar dat hij die gegevens ook kan muteren. Ook bij een datawarehouse heeft hij een omgeving waarin hij het integrale klantbeeld kan vormen. Om in dat geval gegevens te wijzigen moet hij echter de onderliggende systemen benaderen.
Bij een klantdatabase en een interfaceomgeving zijn de consequenties voor de gebruiker afhankelijk van de functionaliteiten die hem ter beschikking staan. Van belang is hierbij hoeveel systemen een gebruiker moet toepassen.
Een belangrijk punt van overweging is ook dat het datawarehouse (in tegenstelling tot de overige architecturen) bij uitstek geschikt is voor het uitvoeren van analyses op geaggregeerde klantgegevens. Daarnaast beschikt een datawarehouse over actuele en historische gegevens, die noodzakelijk zijn om een breed en integraal klantbeeld te kunnen vormen. Hierdoor ondersteunt een datawarehouse ook andersoortige informatiebehoeften (over klantgroepen).

Afweging 3: complexiteit van gegevensuitwisseling

In de regel zijn bij het uitwisselen van gegevens nogal wat conversieslagen nodig. Dat komt doordat veel systemen hun eigen taal spreken. Die taal wordt gevormd door zaken zoals de gebruikte syntax, spellingswijzen (hoe gaan we om met hoofdletters), inkortingsregels (wat te doen als de klantnaam zestig karakters lang is, terwijl er in het systeem maar vijftig beschikbaar zijn) en invulinstructies (wat te doen met het verplichte veld 'geboortedatum', wanneer die datum onbekend is). Dat alles staat nog los van de toegepaste definities. Wanneer gegevens worden uitgewisseld, moeten ze worden vertaald. Niet alleen de complexiteit, maar ook de kosten van het beheer van die uitwisseling worden structureel onderschat.
De architecturen verschillen in de hoeveelheid werk die ze meebrengen op het vlak van gegevensuitwisseling. Het is bijvoorbeeld inmiddels veel organisaties duidelijk geworden dat het inrichten van een kopieerproces voor een datawarehouse veel complexer is dan het opzetten van een aantal interfaces. Dat komt doordat bij een datawarehouse niet alleen de gegevens vertaald moeten worden, maar met name ook doordat historie wordt bewaard. Daarnaast moeten de gegevens uit de verschillende bronnen consistent worden gemaakt. Het kopieerproces levert een extra complicatie op als van buiten de organisatie data over mogelijke klanten binnenkomen. De taal van die externe systemen is door de organisatie veel moeilijker en trager te beïnvloeden.
Een essentieel verschil tussen de eerste drie architecturen heeft betrekking op het aantal malen dat een bepaalde gegevenssoort (in dit geval klantgegevens) wordt opgeslagen. Alhoewel daar nogal eens luchtig over wordt gedaan, zijn wij van mening dat je dit probleem niet te licht moet opvatten. Hoe vaker je een bepaalde gegevenssoort opslaat, hoe meer er wordt uitgewisseld en hoe groter de problemen met consistentie en beheer zijn.
Verder is een belangrijk verschil tussen de architecturen de mate waarin ze aanpassing van de onderliggende systemen noodzakelijk maken. Dat is een complicerende factor, die met name tot veel werk leidt bij het realiseren van een klantendatabase. Een datawarehouse heeft op dit punt de minste impact doordat de onderliggende systemen niet hoeven te wijzigen.

Afweging 4: complexiteit van beheer

Op het gebied van beheer valt er weinig goed nieuws te vertellen. De beschreven architecturen zijn stuk voor stuk complex, wat inhoudt dat ook het beheer ervan complex is. Als we een positieve uitschieter op dit vlak moeten noemen, dan is het wel dat een klantdatabase het best schaalbaar is.

Praktijkinvulling

Gepresenteerd zijn vier ideaaltypische architecturen. In praktijksituaties is het veelal niet zo dat één architectuur de absolute voorkeur geniet. Ze zijn niet eens altijd naar believen in te voeren. In een ERP-systeem kun je bijvoorbeeld niet de klantgegevens zomaar weghalen en vervangen door een *view* naar een andere database. Combinaties van twee, drie en in enkele gevallen zelfs vier architecturen moeten dan ook worden ingezet. Neem bijvoorbeeld een omgeving waarin één bestaand systeem als klantsysteem wordt aangewezen. De soorten klantgegevens die daarin worden opgeslagen, worden uitgebreid, terwijl klantgegevens uit andere systemen worden vervangen door verwijzingen naar het klantsysteem. Voor trendanalyse en managementinformatie wordt een datawarehouse ingezet, terwijl voor het call center een schil wordt ontworpen.

Bewust kiezen

Op zich is het naast elkaar bestaan van oplossingen geen probleem, zolang ze maar het resultaat van een bewuste keuze zijn. Een optimale omgeving ontstaat immers niet zo maar. Zij ontstaat door het nemen van enkele bewuste architectuurkeuzes. Wanneer die keuzes uitblijven, voegen de eigenaren van de verschillende systemen steeds meer functionaliteiten aan hun systemen toe. Daarbovenop moeten steeds meer interfaces gebouwd worden. Wat ons opvalt is dat het omgekeerde, systemen die minder functionaliteiten ondersteunen en minder gegevens vastleggen, minder vaak voorkomt. Een organisatie kan ook overgaan tot het aankopen van een CRM-pakket. Dat pakket wordt ingezet als een klantendatabase, terwijl de daarin benodigde gegevens met interfaces worden gekopieerd uit andere systemen. In dat geval maakt de organisatie impliciet een keus en die is niet per definitie optimaal.

Uitdagingen voor gegevensmanagement

Welke architectuur we ook kiezen, één uitdaging moeten we altijd beslechten. We moeten kunnen bepalen of een klant in het ene systeem dezelfde is als een klant in een ander systeem. Dat vereist (periodieke) matching van gegevens. Daarnaast spelen ook definities een belangrijke rol. Zo moet duidelijk zijn wie we bedoelen wanneer we het hebben over een (zakelijke) klant: is dat de contactpersoon, de personen die het product daadwerkelijk inzetten of de gehele organisatie?
Een speciaal soort definitie heeft betrekking op referentietabellen (ook wel stamtabellen genoemd). Een referentietabel bevat een

verzameling met gegevens die in meerdere systemen worden toegepast. Denk bijvoorbeeld aan een woonplaatsentabel. Een referentietabel voor woonplaatsen geeft uitsluitsel of een klant gevestigd is in Alphen a/d Rijn, Alphen aan den Rijn of Alphen aan de Rijn. Ook geeft zo'n referentietabel aan of Scheveningen een woonplaats is. Een andere referentietabel is de tabel met daarin alle artikelgegevens (productcodes) van de organisatie. Organisaties die een ERP-systeem gebruiken, beschikken vaak al over zo'n tabel. Veel van die organisaties hebben in de praktijk mogen ervaren hoeveel moeite het kost zo'n tabel te realiseren. Op de keper beschouwd is de klantdatabase eigenlijk (een set van) referentietabel(len).

Wijzigingsbevoegdheden

Een vraag die naar boven komt bij het realiseren van het integraal klantbeeld is: wie mag klantgegevens wijzigen? Plat gezegd komt deze vraag neer op: van wie is een klant? De discussie hierover kan heftig en pijnlijk worden en veel energie kosten. Een middel om de discussie van zijn stekelige kantjes te ontdoen kan zijn de gegevens op te splitsen in persoonsgegevens en rolgegevens. Het aanbrengen van een dergelijke structuur kan de discussie over wie klantgegevens mag wijzigen aanzienlijk versoepelen. In zo'n structuur wordt een onderscheid gemaakt tussen gegevens over (rechts)personen en gegevens over de rollen die zij spelen.
Over de bevoegdheid van het wijzigen van persoonsgegevens zal weinig discussie ontstaan. De mogelijke rollen die een klant vervult, bieden handvatten om te bepalen wie in de organisatie gegevens mag wijzigen die betrekking hebben op die rol. Het modelleren naar rollen maakt het stukken eenvoudiger om te gaan met situaties waarin een natuurlijke persoon bijvoorbeeld nu eens verzekerde, dan weer een rekeninghouder en in weer een andere situatie een eigen werknemer is. Een rechtspersoon kan voor de ene zaak een leverancier en een crediteur zijn en voor de andere een afnemer en debiteur. Ook de samenhang tussen rollen moet worden gemodelleerd. Op die manier kan men bijvoorbeeld ook eenduidig de wijzigingsbevoegdheden toewijzen in het geval dat een eigenaar van een onderneming die als leverancier diensten levert, tevens rekeninghouder is en zijn kinderen als verzekerde geregistreerd heeft staan.

Overige uitdagingen

Een andere mooie uitdaging vloeit voort uit het feit dat in *elk* klantcontactmoment het integrale klantbeeld beschikbaar moet zijn. Dat wil zeggen, in het call center, op de winkelvloer, op het

Internet en voor de accountmanager onderweg. Dat stelt eisen aan de IT-infrastructuur. Immers, de datacommunicatie moet beschikbaar zijn, de gebruikte operatingsystems moeten op elkaar afgestemd zijn en de *security* moet *in place* zijn.
Een laatste uitdaging die wij hier noemen, is dat bij elk klantcontactmoment processen opgestart kunnen worden. De klant kan immers bij elk klantcontactmoment een order plaatsen. Het managen van de afhandeling moet gelijk zijn voor elke order, ongeacht via welk distributiekanaal deze is ingebracht. De voortgang van elke order moet (ook door de klant) te volgen zijn, wederom ongeacht de wijze waarop deze is ingebracht. Dat stelt eisen aan de wijze van organiseren, aan de structuur van de software en aan de workflow om die software aan elkaar te koppelen.

Het realiseren van een adequate architectuur levert ons, IT'ers, nog heel wat werk op. Dat is niets om voor terug te schrikken, maar dwingt wel tot enige behoedzaamheid wanneer afdeling Marketing komt vragen om één integraal klantbeeld. Die behoedzaamheid is nodig om de verschillende problemen in onderlinge samenhang stap voor stap aan te pakken. Uiteraard moeten die stappen wel in lijn zijn met een lange-termijngedachte, die bestaat uit een consistente architectuur.

Auteursgegevens

Drs. ing. Mark van der Pas is Informatiemanager bij Libertel NV.
Drs. Rien Hamers is docent aan de Fontys Hogeschool Informatica te Eindhoven.

Van der Pas en Hamers hebben dit artikel op persoonlijke titel geschreven.

Dit artikel is eerder verschenen in Informatie Management, oktober 2000.