

Op weg naar een Enterprise Application Integrator

# De toekomst voor de Datawarehouse-architectuur

Sytse-Jan van Spronsen

**O**rganisaties zijn in de afgelopen decennia voor hun functioneren steeds vaker afhankelijk geworden van ICT. Het gebruik van de computer is veranderd van een stand alone-apparaat, waarmee in de jaren zeventig met batch-programma's specifieke berekeningen gemaakt werden, naar een conglomeraat van real time-informatie-systemen, waarmee een gehele organisatie wordt ondersteund. In dit artikel over samenhang en integratie staan Datawarehouse en Enterprise Application Integration centraal en wordt een overzicht gegeven van beschikbare toepassingen. Aansluitend zijn huidige eisen aangegeven, waaruit kan worden afgeleid welke architectuur gewenst is.

De ontwikkeling van de ICT heeft in de loop der jaren een aantal stromingen opgeleverd, die meer samenhang tussen de verschillende informatiesystemen proberen te verkrijgen. Aan de ene kant is door ICT-Consultants getracht om via (Strategic) Information System Planning verbeteringen aan te brengen in de functionele en de data-structuur. Dit heeft over het algemeen geleid tot omvangrijke ICT-ontwikkelingstrajecten waarbij, voor zover mogelijk, gebruik is gemaakt van standaardapplicaties aangevuld met maatwerkt toepassingen.

Aan de andere kant hebben applicatieleveranciers met ERP-systemen getracht om kant en klare toepassingen te bieden voor de gehele organisatie. Dit heeft omvangrijke ICT-implementatietrajecten tot gevolg gehad, waarbij slechts de algemene functionaliteit van organisaties werd ondersteund. Naast de ERP-pakketten zijn vaak nog talrijke specifieke applicaties nodig geweest.

## INTERFACES

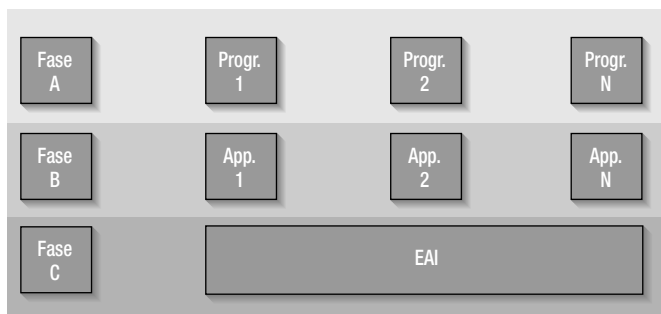
In de praktijk blijkt dat het aantal administratieve toepassingen wel is verminderd, maar dat binnen een organisatie nog steeds een groot aantal applicaties wordt gebruikt. In diverse situaties wordt daarom niet gestreefd naar een vermindering van het aantal applicaties, maar wordt getracht de applicaties te integreren. Enerzijds komt integratie voort uit het feit dat het management inzicht in het functioneren van de organisatie wil hebben en rapportages van

gegevens, die in meerdere applicatiedatabases zijn vastgelegd. Over het algemeen heeft dit geleid tot een Datawarehouse (DW) architectuur. Op basis van deze gegevens kunnen management-rapportages vanuit meerdere gezichtspunten eenvoudig worden opgesteld.

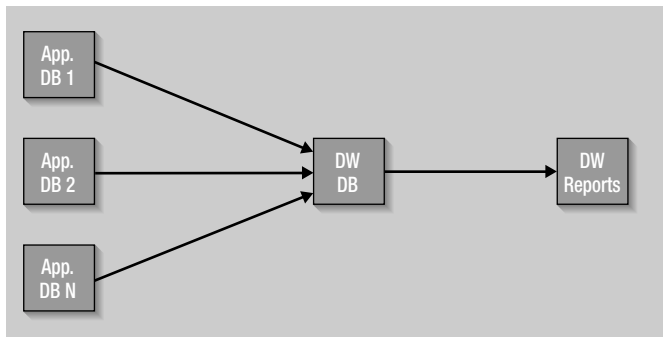
Anderzijds is een integratie-architectuur ontstaan door de opkomst van internet, waardoor organisaties vaak genoodzaakt werden tot het aanschaffen of vervaardigen van Internet Front Office toepassingen. Het afstemmen van deze toepassingen op organisatie en back office-systemen heeft veelal tot gevolg dat een groot aantal extra handmatige handelingen of interfaces tussen systemen nodig is. Zo dienen eenvoudige gegevens, zoals NAW of orderinformatie vaak twee keer in verschillende systemen te worden vastgelegd. Uit oogpunt van doelmatigheid heeft dit geresulteerd in een behoefte naar data entry-integratie van de verschillende systemen. Dit type integratie staat tegenwoordig bekend als Enterprise Application Integration (EAI). Figuur 1 geeft een overzicht van de ontwikkeling van stand alone-programmatuur naar EAI.

## DW-PROGRAMMA'S EN TOOLS

Voor zowel DW als EAI zijn inmiddels veel tools en hulpmiddelen beschikbaar, en die zijn in hun opzet niet altijd even duidelijk. Een DW is te definiëren als een centrale database, waarin gegevens vanuit meerdere applicatiedatabases zijn geïntegreerd ten behoeve van (management) rapportages. Hierbij zijn de benodigde tools in de volgende onderdelen op te splitsen:



FIGUUR 1: GLOBALE FASERING ONTWIKKELING ICT-ARCHITECTUUR.



FIGUUR 2: STANDAARD DW-ARCHITECTUUR.

1. De centrale database waarin de gegevens worden opgeslagen
2. Data extractie-tools, waarmee de gegevens vanuit de applicatiedatabases kunnen worden overgezet naar de centrale database.
3. Rapportage-tools waarmee de gegevens in de centrale database kunnen worden geanalyseerd.

Voor het centrale DW, waarin de historische gegevens vanuit de verschillende applicatiedatabases worden verzameld, wordt veelal gekozen voor een van de bekende relationele (SQL) databases, om een grotere keuzevrijheid te hebben ten aanzien van de benodigde conversietools en rapportgeneratoren.

In verkoopbrochures van DW-tools wordt vaak gesuggereerd dat men met een DW over een tool beschikt, waarmee op vrijwel elke managementvraag direct een antwoord kan worden gegeven.

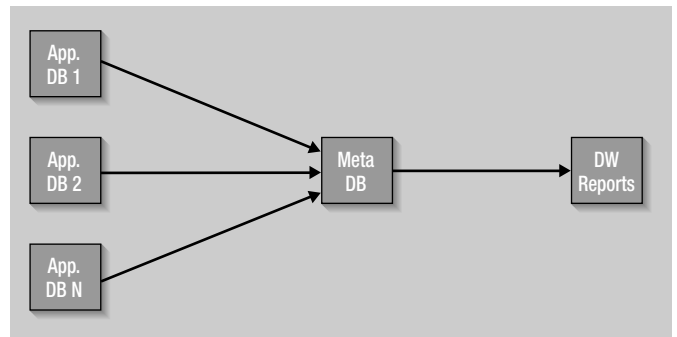
### Een integratie-architectuur is ontstaan door de opkomst van internet

Helaas dient voordat het zover is eerst een ontwerp voor de database te worden gemaakt op basis van de rapportagewensen. Afhankelijk van de opgenomen gegevens kan dus bepaalde rapportage wel of niet worden vervaardigd. Zijn bijvoorbeeld geen financiële gegevens van projecten in een DW beschikbaar dan zal het ook niet mogelijk zijn projectresultaten per relatie te tonen.

Bij het definiëren kunnen de bouwers en ontwerpers nog op enkele praktische problemen stuiten, zoals:

1. Het ontbreken van een eenduidige datadefinitie waarmee het mogelijk is gegevens te integreren vanuit verschillende applicaties. Zo kunnen in de afzonderlijke applicaties relatienummers op verschillende wijzen zijn vastgelegd. Een debiteur is in een grootboekprogramma vaak op andere wijze vastgelegd, dan hij als client in het CRM-pakket vermeld staat. Toch is het voor een rapportage gewenst dat de gegevens van eenzelfde debiteur kunnen worden gematched binnen het DW.
2. Het niet direct beschikbaar hebben van de gegevens die hard coded zijn opgenomen in de applicatieprogrammatuur. De programma's kunnen bijvoorbeeld eenvoudige tabellen die gebruikt worden voor verificatie van de invoer, hardcoded bevatten.

Het zal duidelijk zijn dat ontwerpen en inrichten van een DW



FIGUUR 3: REALTIME DW-ARCHITECTUUR.

dient te worden opgevat als een gewoon ICT-ontwikkelingsproject met de daarbij behorende doorlooptijd en kosten.

Voor het verzamelen van de gegevens uit de verschillende applicatiedatabases, is het noodzakelijk dat de conversies en extractie periodiek als batchproces worden uitgevoerd. Vervolgens kunnen de gewenste overzichten opnieuw worden opgevraagd. De standaardarchitectuur van een DW is weergegeven in figuur 2. Een nieuwe trend is dat het DW (near) realtime gevoed kan worden vanuit de bronsystemen. Voor het maken van (near) realtime-extracties kan gekozen worden voor twee type architecturen, namelijk de standaard DW-architectuur waarbij het DW near realtime wordt gevoed of een architectuur waarbij de rapportages rechtstreeks worden afgeleid vanuit de applicatiedatabases. De laatste architectuur heeft het voordeel dat geen specifieke database voor het DW ontworpen hoeft te worden. Wel is het nodig dat in een meta-database wordt vastgelegd op welke wijze de gegevens vanuit de verschillende applicatiedatabases met elkaar kunnen worden gekoppeld, zie figuur 3.

Andere specifieke tools voor DW, zoals datamart of datamining worden binnen het kader van data-integratie buiten beschouwing gelaten.

### EAI-PROGRAMMA'S EN -TOOLS

EAI-tools hebben de integratie van applicaties als doelstelling. Zo is het gewenst dat de invoer van gegevens eenmalig plaats vindt, bijvoorbeeld een organisatie die wordt vastgelegd in een CRM-pakket wordt ook direct vastgelegd als debiteur in een financieel pakket. Daarnaast is het gewenst dat overzichten kunnen worden verkregen uit de databases van verschillende applicaties. Is bij het DW sprake van een eenduidige architectuur, zie figuur 1, bij EAI-tools is dit zeker niet het geval.

Wat betreft de opzet is een onderscheid te maken in de volgende basisarchitecturen, waarbij de integratie op verschillende niveau's aansluit:

1. User Interface-niveau; Screen Scrapers  
Bestaande applicaties kunnen worden geïntegreerd door op user interface-niveau, gegevens van de ene applicatie op te vangen en deze door te geven aan de andere. De veranderingen voor de gebruiker zijn op applicatieniveau beperkt. Er

wordt namelijk gebruik gemaakt van dezelfde toepassingen. Om de verwerking van gegevens te bewaken, is ondersteuning nodig met een workflow-laag.

2. Applicatieniveau; API's, Applicatie Adaptors

Integratie op applicatieniveau maakt gebruik van specifieke interfaces per applicatie, die beschikbaar zijn gesteld als API's (Application Program Interface) of Application Adaptors.

Integratie op het laagste niveau integreert de gegevens op databaseniveau.

3. Databaseniveau; Conversie tools

Deze laatste basis-architectuurvorm werkt de gegevens van de verschillende applicaties rechtstreeks bij in diverse applicatie-databases.

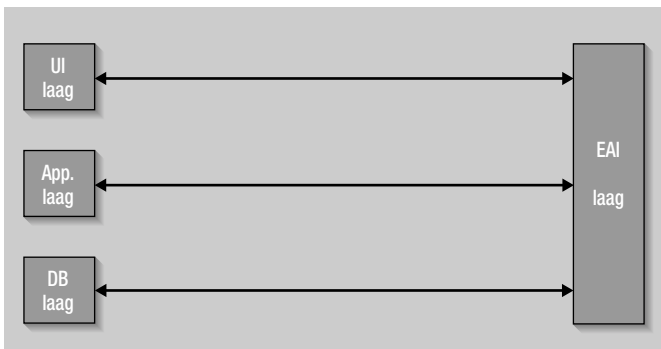
Integratie kan geregeld worden tussen de betrokken applicaties onderling of kan worden ondersteund met een afzonderlijke bovenlaag, zie figuur 4. In dit laatste geval is het mogelijk, zeker als er een groot aantal applicaties geïntegreerd dient te worden, veel minder koppelingen te realiseren. In de centrale bovenlaag zijn oplossingen mogelijk die gebruik maken van standaardisering van de interfaces, bijvoorbeeld via een centrale XML-database, of door gebruik te maken van een meta-database met daarin opgenomen de business-logica waarmee de integratie kan worden gerealiseerd.

Het zal duidelijk zijn dat een organisatie eerst een keuze moet maken voor de gewenste integratie-architectuur, inclusief de daarvoor beschikbare programmatuur en tools. De nadruk bij EAI ligt veelal op het bieden van (near) realtime Web-oplossingen, met een integratie van traditionele back office-applicaties. Om de problematiek in dit artikel niet complexer te maken is de benodigde communicatie-middleware buiten beschouwing gelaten.

Als gekozen is voor een specifieke integratie-architectuur, kan met het integreren van de verschillende applicaties worden gestart, op vergelijkbare wijze als de realisatie van een 'gewoon' ICT-project.

**INTEGRATIEWENSEN EN -EISEN**

Het bepalen van een applicatie-integratie-architectuur is voor een organisatie door het hoge abstractieniveau moeilijk af te leiden uit de wensen en eisen van de organisatie zelf. Wel kan een aantal algemene wensen en eisen worden onderkend die helpen bij het



FIGUUR 4: BASIS EAI-ARCHITECTUUR.

bepalen van de toekomstige integratie-architectuur. Dit zijn achter-eenvolgens;

**Realtime verwerking**

Bij het vastleggen van gegevens ligt het voor vrijwel iedereen vast dat de gegevens ook direct in eventueel verschillende administraties worden verwerkt. Plaatst u bijvoorbeeld een overboeking van uw rekening courant naar uw internetspaar-rekening dan wilt u natuurlijk direct het actuele saldo op uw rekening courant inclusief uw bankafschriften kunnen opvragen.

Voor de architectuur betekent dit, dat bij het (Near) Realtime verwerken bij voorkeur het aantal extra stappen beperkt dient te worden gehouden. Bij data-entry worden daarom gegevens bij voorkeur direct verwerkt in de losstaande applicatiedatabases en voor het opstellen van rapporten wordt indien mogelijk gewerkt zonder een afzonderlijk DW, dat afzonderlijk gesynchroniseerd dient te worden.

**Multi-language ondersteuning**

Het beschikbaar stellen van toepassingen aan klanten via het internet dwingt een organisatie tot het opzetten van deze toepassingen in verschillende talen. Een (potentiële) klant wil graag de beschrijvingen van producten of diensten lezen in de eigen taal en en wil z'n eventuele bestelling ook kunnen boeken in die taal.

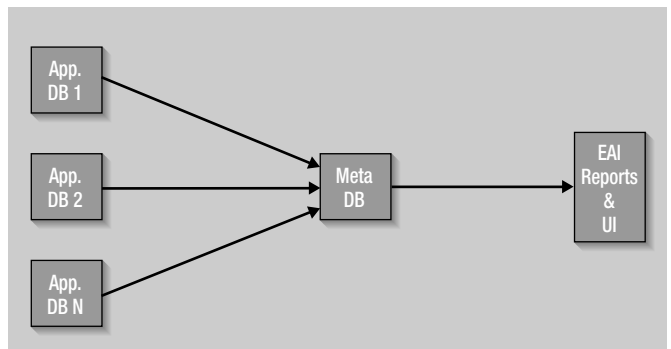
*Zo dienen eenvoudige gegevens vaak twee keer te worden vastgelegd*

Het is dus wenselijk dat bij het opzetten van nieuwe applicaties gebruik gemaakt kan worden van een opzet waarmee het niet nodig is om voor elke taal een nieuwe applicatie te gaan ontwikkelen.

Applicaties die over Multi-language functionaliteit beschikken hebben de user interface vaak losgekoppeld van de gegevensverwerking. De omschrijving en lay-out wordt dan per taal vaak vastgelegd in een meta-database. Deze meta-database kan worden toegepast voor zowel de UI-definitie als voor de Rapportages. De ideale integratie-architectuur is dan ook zo opgezet.

**Huisstijl**

Het beschikbaar stellen van verschillende internettoepassingen



FIGUUR 5: DATABASE-GEÖRIENTEERDE EAI-ARCHITECTUUR.

aan (potentiële) klanten dwingt uit het oogpunt van eenvoud en eenduidigheid tot een uniforme presentatie. Het zal voor een klant erg verwarrend overkomen als zijn bankrekeningafschrift, dat hij via internet opvraagt niet gelijk is aan het rekeningafschrift dat via de post thuis wordt ontvangen. Lay-out van zowel de UI als van rapportages zal afgestemd moeten worden op de huisstijl van een organisatie.

Met betrekking tot de architectuur, is een loskoppeling van de lay-out met de gegevensverwerking gewenst. Zo is het gewenst dat bij het definiëren en wijzigen van invoerschermen of rapporten

ties, betekenen meestal dat extra gegevens vastgelegd dienen te worden. Zowel de applicatiedatabase als de UI en rapportages dienen te worden aangepast.

Wordt gebruik gemaakt van integratie op UI-niveau, dan zal voor elk van de schermen specifiek weer een integratie moeten worden gemaakt. Het opnemen van de extra functionaliteit op applicatieniveau houdt waarschijnlijk in, dat het nodig is om extra API's op te nemen of dat nieuwe Applicatie Adapters moeten worden gemaakt. Het toevoegen van nieuwe gegevens en functies in een geïntegreerde applicatie-omgeving is verreweg het eenvoudigst te realiseren bij gebruik van data-integratie. Nieuwe tabellen en velden kunnen eenvoudig worden toegevoegd aan een database en de integratiesoftware dient vervolgens de mogelijkheid te bieden om een UI te maken voor het benaderen van deze gegevens en het verzorgen van de rapportages.

**Een organisatie moet eerst een keuze maken voor de gewenste integratie-architectuur**

gebruik gemaakt kan worden van standaard-templates die beschikbaar zijn in een centrale bibliotheek of meta-database.

**Flexibiliteit**

De snelheid waarmee een organisatie nieuwe producten of diensten op de markt kan brengen is vaak van cruciaal belang. Wie als eerste potentiële klanten kan overtuigen van een nieuw product of dienst, heeft een groot voordeel. Weet een organisatie hiermee de groep van early adaptors als eerste te overtuigen, dan levert dit succes een voordeel voor zowel de naamsbekendheid als voor eventuele leereffecten waarmee de voorsprong in de markt kan worden vergroot. De druk om sneller toepassingen te kunnen aanpassen zal dan ook steeds groter worden. Marketeers denken hooguit aan een introductietijd van enkele maanden in plaats van de enkele jaren, die het nu kost bij de oplevering van een nieuwe of aangepaste applicatie.

Flexibiliteit kan bij integratie projecten worden onderverdeeld in:

- 1. Het toevoegen van nieuwe applicaties.

Wordt een nieuwe applicatie toegevoegd in een integratieproject, dan is het handig dat slechts een eenmalige koppeling tot stand moet worden gebracht in plaats van koppeling naar elk van de systemen. Een integratie-architectuur met een centrale laag is dan ook te prefereren.

- 2. Het wijzigen van bestaande applicaties.

De extra wensen die worden toegevoegd aan bestaande applica-

**CONCLUSIES**

De ICT-markt heeft zich in de afgelopen jaren ontwikkeld van gebruik van specifieke programma's ter ondersteuning van bepaalde berekeningen naar geïntegreerde toepassingen. In de eerste plaats heeft de integratie zich voorgedaan bij het DW (figuur 2) en zich meer in de richting van (near) realtime rapportagesystemen (figuur 3) ontwikkeld. De integratie bij user interfaces is met name door de internetapplicaties gestimuleerd. De integratie-applicaties op data entry-niveau zijn te onderscheiden in integratie op UI-niveau, Applicatie-niveau en Database-niveau (figuur 4).

Een te verwachten ontwikkeling is dat de integratie op database-niveau zal plaats vinden, met gebruikmaking van een meta-database ter ondersteuning van benodigde datalinks en uniforme presentaties (figuur 5).

Er is in dit artikel een aantal eisen en wensen op een rij gezet en aangegeven in hoeverre deze eisen en wensen worden ondersteund door de genoemde integratie-architecturen, (figuur 6). Het blijkt dat de database-geïntegreerde EAI-architectuur als enige positief opvalt. In de toekomst zal de architectuur van het Datawarehouse daar naar toe groeien.

Drs. Ing. Sytse-Jan van Spronsen (sytse@iquad.nl) werkt als senior Business Consultant bij Iquad Consultancy B.V.

	Architectuur	2	3	4	5
<b>Wensen / Eisen</b>					
1	(near) Realtime	-	+	+	+
2	Multi-Language	-	-	-	+
3	Huisstijl UI	-	-	-	+
4	Toevoegen van Applicaties	-	0	-	0
5	Wijzigen van Applicaties	-	+	-	+
6	Standaard Rapportages	+	+	-	+
7	Data Entry Support	-	-	-	+

**FIGUUR 6: INTEGRATIE ARCHITECTUUR EN WENSEN/EISEN.**