

## Datamodellering / Informatie-analyse volgens NIAM

### code/duur/doelgroep/leerdoelen

*code*  
IAN

*duur*  
2 dagen

#### *prijs, data en inschrijven*

Op de pagina Datamodellering / Informatie-analyse volgens NIAM op de website van ADVICOM staan de verdere details over wanneer de cursus / workshop wordt gehouden, wat de prijs is, en hoe je kunt inschrijven: [www.advicom.nl/HTML/cursussen/ian.html](http://www.advicom.nl/HTML/cursussen/ian.html)

#### *doelgroep*

informatie-analysten, systeemontwikkelaars, database ontwerpers, functioneel beheerders, database beheerders, data warehouse ontwikkelaars, opstellers van requirements en verder iedereen die in zijn of haar werk met datamodellen te maken krijgt

#### *leerdoelen*

De deelnemer is in staat:

- een datamodel / informatiemodel op te stellen gebruik makend van de methode NIAM
- dingsoorten (entiteitstypen) te kiezen die bedrijfsbreed bruikbaar zijn, stabiel zijn en vanuit administratief technisch oogpunt te volgen zijn
- datamodellen / informatiemodellen zo op te stellen dat ze praktisch bruikbaar zijn
- om te gaan met algemene datamodellerings- / informatiemodelleringsconcepten
- een NIAM-datamodel / NIAM-informatiemodel om te zetten in een relationeel gegevensmodel in BCNF

### onderwerpen

In de workshop komen de volgende onderwerpen aan de orde:

- kaderplaatsing datamodellering- / informatiemodellering
- aspecten van informatie
- kiezen van dingsoorten (entiteitstypen) en feittypen (associatietypen)
- NIAM-werkwijze
- n-aire feittypen
- verwoorden, classificeren, kwalificeren
- nominalisatie
- overtolligheid
- dingsoort versus naamsoort
- informatieregels

- tijdsaspecten
- complexe structuren
- omzetten datamodel / informatiemodel in relationeel gegevensmodel

### algemeen

#### *inleiding*

Het maken van een datamodel / informatiemodel in het kader van het ontwikkelen van informatiesystemen is een essentiële activiteit. In de praktijk komt het helaas veel te vaak voor dat men het nut van een goede analyse met als resultaat een datamodel / informatiemodel niet inziet. Het lijkt immers zo eenvoudig. Kom je tot ontdekking bij het bouwen van een informatiesysteem dat je gegevens moet opslaan, dan maak je daarvoor toch gewoon een tabelletje.

Deze gedachte is begrijpelijk. De techniek maakt het ook heel eenvoudig te realiseren. Wat hierbij uit het oog verloren wordt is dat zo'n tabelletje wel goed zal werken voor de applicatie die men aan het ontwikkelen is, maar dat het maar de vraag is of het tabelletje ook te gebruiken is in combinatie met andere applicaties.

Om vervolgens het gebruik van de tabel in combinatie met een nieuwe applicatie mogelijk te maken, wordt vaak een aparte tabel opgenomen waarin voor de nieuwe applicatie de uitzonderingen staan die de oude applicatie niet kent en die de oude applicatie ook niet in de oorspronkelijke tabel mag tegenkomen.

Dit maakt de programmatuur van de nieuwe applicatie nodeloos complex. Het is bovendien moeilijk management informatie uit deze tabellen te halen. In de praktijk stapelt men vaak uitzonderingstabel op uitzonderingstabel met als gevolg dat programma's slecht werken. De database wordt daardoor onbetrouwbaar en de programmatuur is moeilijk te onderhouden.

Dit probleem en nog een heleboel andere problemen kun je vermijden door het opstellen van een datamodel / informatiemodel. De reden is dat een datamodel / informatiemodel is opgebouwd uit elementen (dingsoorten, naamsoorten en feittypen) die getoetst worden op hun bruikbaarheid voor de gehele organisatie (en daarmee voor meer dan één applicatie).

#### *concepten*

De analist moet goed thuis zijn in de concepten achter het modelleren. Bij concepten moet men denken aan bijvoorbeeld het juist kiezen van dingsoorten, het omgaan met allerlei structuurvormen, aandragen van structuren die een bijdrage geven aan het oplossen van het pro-

## Datamodellering / Informatie-analyse volgens NIAM vervolg

bleem waarvoor de applicatie gebouwd wordt.

In de workshop ligt veel nadruk op het beheersen van de modelleringsconcepten.

### *omgaan met complexiteit*

Een datamodel / informatiemodel kan behoorlijk complex worden. Dit heeft allerlei oorzaken. Het behandelde probleem kan bijvoorbeeld complex zijn, zoals in planningssystemen, waarbij diverse planningen elkaar beïnvloeden, waar planningen in meer fasen tot stand komen, afwijkingen geanalyseerd moeten kunnen worden, enzovoort. Andere voorbeelden van oorzaken die complexiteit in het datamodel / informatiemodel veroorzaken zijn:

- tijdsaspecten
- dingsoorten die in diverse rollen voorkomen
- interfaces met andere systemen
- dingsoorten die niet (of moeilijk) administratief te volgen zijn

In de workshop wordt behandeld hoe de deelnemer met complexiteit kan omgaan.

### *omzetten datamodel / informatiemodel in een relationeel gegevensmodel*

De belangrijkste reden waarom datamodellen / informatiemodellen worden opgesteld is het gebruik ervan bij het opzetten van een tabellenstructuur voor een database. Iedere data- / informatiemodelleringsmethode heeft daartoe een stappenplan, waarmee een datamodel / informatiemodel kan worden omgezet in een gegevensstructuur volgens het relationele model (geschikt als ontwerp van de tabellen in een relationele database). Het toepassen hiervan is één van de onderwerpen van de workshop.

### *informatieregels*

Belangrijk onderdeel van een datamodel / informatiemodel is de specificatie van informatieregels (in NIAM beperkingsregels genoemd). Informatieregels worden gebruikt voor het controleren van informatie op het voldoen aan gestelde regels (juistheid). Daarnaast worden informatieregels gebruikt voor het specificeren van functionaliteit van het te bouwen informatiesysteem.

Wij zullen dit duidelijk maken aan de hand van een drietal voorbeelden:

- bij het invoeren van een nieuw huwelijk in het huwelijksregister van een gemeente wordt door middel van een informatieregel gecontroleerd of de betreffende partners geen andere op het geplande moment van huwen geldige huwelijken hebben. Hiermee wordt er voor gezorgd dat ingevoerde feiten voldoen aan de gestelde regels

- een effectenkantoor heeft zo nu en dan de preciese schuldpositie met een client nodig. Deze schuldpositie is via een informatieregel af te leiden. De informatieregel (een voorbeeld van een afleidingsregel) beschrijft hoe de schuldpositie is af te leiden uit transacties die sinds de vorige bepaling van de schuldpositie zijn opgenomen. De regel verrekent daarbij provisie, rente enzovoort. In het te bouwen systeem wordt software (functionaliteit) opgenomen om de schuldpositie uit te rekenen. In de database zal de schuldpositie niet actueel worden bijgehouden. Dit komt omdat de positie slecht eens per zoveel transacties benodigd is en het rekenwerk gigantisch is. Het actueel bijhouden zou betekenen dat cijfers berekend worden die nooit geraadpleegd worden. Dat is niet zinvol

- een rederij moet regelmatig een bemanning samenstellen voor de schepen. Dit gebeurt aan de hand van een informatieregel. In deze regel wordt rekening gehouden met beschikbaarheid (andere planningen, ziekte, vrije dagen en vakantie) van het personeel. Verder gebeurt dit aan de hand van bekwaamheden, voorkeuren, optimale inzet over perioden heen, kostprijs, enzovoort. De uitkomst is een volledige bemanning. Ook voor deze regel wordt een stuk software ontwikkeld dat onderdeel is van de functionaliteit van het te ontwikkelen systeem. Uit deze voorbeelden blijkt dat een datamodel / informatiemodel met een volledige specificatie van informatieregels (business rules) voor meer te gebruiken is dan alleen voor het opzetten van een database. Het datamodel / informatiemodel vormt de basis voor de volledige specificatie van het te bouwen informatiesysteem. Dus ook voor de functionaliteit.

In de workshop leert de deelnemer met informatieregels om te gaan.