

De unificatie van objecttypen en feittypen

een praktisch en didactisch vruchtbare theorie

H. van der Lek, G.P. Bakema en J.P.C. Zwart

In moderne feittype-georiënteerde methoden voor informatie-analyse, zoals n-air nested NIAM (Nijssens Informatie Analyse Methode), wordt gebruik gemaakt van de mogelijkheid feiten te zien als objecten. Voor dit objectificeren van feiten, dat evenals het feitbegrip zelf een taalkundige grondslag heeft, gebruikt Nijssen ook wel de linguïstische term nominalisatie. In dit artikel wordt ook het objectbegrip voorzien van een betere linguïstische grondslag door alle niet-lexicale objecten in NIAM consequent te beschouwen als nominalisaties van feiten. Wij zien deze stap als de logische consequentie van de filosofie dat informatiemodellering niet de werkelijkheid zelf, maar de communicatie over de werkelijkheid beoogt te modelleren (communication oriented approach). Een belangrijk theoretisch en praktisch resultaat van deze zienswijze is een eenvoudige architectuur voor CASE-tools voor het vastleggen van NIAM-feitengrammatica's en het genereren van relationele schema's daaruit via database-operaties. Van didactisch belang is het simpele generieke gegevensmodel waartoe deze zienswijze leidt en waarvan niet alleen NIAM en het relationele model speciale gevallen zijn, maar ook andere informatiemodellen, zoals het binaire, het functionele, het semantische en het entity-relationship-model, die eruit ontstaan door het opleggen van daartoe geëigende beperkingsregels.

1 Inleiding en probleemstelling

Het gebruik van en het onderwijs in NIAM (Nijssens Informatie Analyse Methode) is de laatste jaren sterk gegroeid. Het boek van Wintraecken (Wintraecken, 1985) heeft daartoe ook bijgedragen. Men vindt daarin echter slechts de binaire variant van NIAM, waarop inmiddels enkele CASE-tools zijn gebaseerd. In het boek van Nijssen en Halpin (Nijssen & Halpin, 1989) wordt het veel krachtiger n-air nested NIAM beschreven (al eerder (Nijssen, 1977) bestond de idee van meer dan binaire feittypen). Daarin worden ook feittypen met meer dan twee rollen toegestaan en wordt de mogelijkheid geboden feittypen onder bepaalde condities te zien als objecttypen, die zelf weer rollen spelen in andere feittypen (nesten van feittypen). Dit tot objecttypen maken van feittypen (Nijssen & Halpin, 1989, blz. 54) wordt naar de taalkundige basis ervan door Nijssen ook wel nominalisatie genoemd (Nijssen, 1990, blz. 134).

De eerste aanleiding voor de in dit artikel gepresenteerde gedachten was de behoefte aan een op n-air nested NIAM gebaseerd CASE-tool voor het vastleggen van feitengrammatica's en het daaruit genereren van relationele schema's.

De vaststelling dat een gegevensbank geen representaties van objecten uit de werkelijkheid bevat (object oriented view), maar slechts feiten waarin verwijzingen (labels zoals namen, nummers, codes) naar die objecten een rol spelen (fact oriented approach; zie Nijssen & Halpin, 1989), vormt het centrale principe van NIAM. Anders gezegd: informatiemodellering beoogt niet de werkelijkheid zelf te modelleren, maar de communicatie over de werkelijkheid. In Wintraecken (1985) en Nijssen & Halpin (1989; zie bijvoorbeeld blz. 49) wordt echter in de fundering van het feitbegrip in wezen toch gebruik gemaakt van (pseudo-)objecten in plaats van hun labels in de bevolking van alle rollen die gespeeld worden door niet-lexicale objecttypen (entiteittypen). Dit is strijdig met bovenvermeld principe.

De wens deze (pseudo-)objecten fundamenteel uit te bannen vormde de tweede meer theoretische aanleiding voor onderzoek naar de grondslagen van NIAM, vanuit een consequente communication oriented approach, zoals wij deze benadering noemen.

Voor het informatieperspectief van informatiesystemen bestaan er slechts drie lagen (Nijssen & Halpin, 1989; Nijssen, 1990). De onderste laag of feitenbank (FB) bevat alle gebruikersfeiten (denk aan de bevolking van een NIAM-grammatica of aan de inhoud van de tabellen van een relationele database). De middelste laag of feitengrammatica (FG) is een precieze specificatie van de toegestane feiten in de FB in termen van feittypen en beperkingsregels (denk aan een NIAM-diagram of aan een relationeel schema, waarin de belangrijkste elementen van een NIAM-diagram worden weergegeven). De bovenste laag of metagrammatica (MG) is dan tenslotte een precies voorschrift voor de vorm en de beperkingsregels van FG's. Met andere woorden, de MG bepaalt hoe een FG eruit mag zien en omdat de MG zelf ook een feitengrammatica is, bepaalt zij ook hoe zij er zelf uitziet en bestaan er geen hogere lagen. Een MG is echter per gegevens-/informatiemodel (het relationele model, het entity-relationship-model, NIAM enzovoort) verschillend, hoewel er vaak veel overeenkomsten bestaan tussen al deze modelspecifieke MG's.

Vanuit zowel theoretisch als didactisch oogpunt ontstond de behoefte aan een generiek gegevensmodel, geconcretiseerd in een generieke MG, waarin deze modelafhankelijkheid wordt opgeheven, zodat er een gemeenschappelijke zienswijze ontstaat op de verschillende gegevens-/informatiemodellen.

Door het invoeren van één enkel uitgangspunt is het mogelijk aan de drie genoemde wensen te voldoen. Dit principe, dat ook als titel voor dit artikel is gekozen, luidt:

Alle niet-lexicale objecttypen zijn nominalisaties van feittypen.

In paragraaf 2 worden de principes aan de hand van voorbeelden aanschouwelijk gemaakt. In paragraaf 3 laten we zien hoe dit leidt tot een MG voor de architectuur van een CASE-tool voor het vastleggen van NIAM FG's en het daaruit via database-operaties genereren van ONF relationele schema's. In paragraaf 4 tonen we in essentie hoe deze gegevensmodel-onafhankelijke MG door het opleggen van modelspecifieke beperkingsregels overgaat in gegevensmodel-afhankelijke MG's. In paragraaf 5 tenslotte worden enkele opmerkingen gewijd aan de praktijk, de theorie en het onderwijs.

N.B. Iedere methode voor informatiesysteemontwikkeling (ISDM) heeft een methodologische kant (de voorschriften hoe te komen tot een resultaat) en een structureel aspect (de onderliggende filosofie en de structuur van het resultaat zelf). In dit artikel beperken we ons voornamelijk tot het structurele aspect en daarbij dan nog uitsluitend tot het informatieperspectief. Methodologisch komt slechts het hoognodige aan de orde (zie daarvoor Nijssen & Halpin, 1989).

2 Objecten en feiten

In paragraaf 2.1 behandelen we vanuit de communicatie-georiënteerde zienswijze in vogelvlucht de belangrijkste NIAM-begrippen (zin, zinstype, feit, feittype, object, objecttype, label, labeltype, nominalisatie, nesting van feittypen en beperkingsregels) en de voor NIAM nieuwe begrippen objectformulering en objecttypeformulering. In paragraaf 2.2 laten we zien dat het consequent doorvoeren van deze aanpak leidt tot equivalentie van feittypen en objecttypen.

2.1 Taalkundige grondslagen

De eerste stappen van de analyse volgens NIAM zijn (Nijssen & Halpin, 1989; Nijssen, 1990):
50 Verzamel of stel op: concrete voorbeelden van voor de communicatiepartners (gebruikers) vertrouwde feiten over zekere voor hen relevante werkelijkheid (het communicatiedomein, in het Engels: Universe of Discourse, afgekort tot UoD).

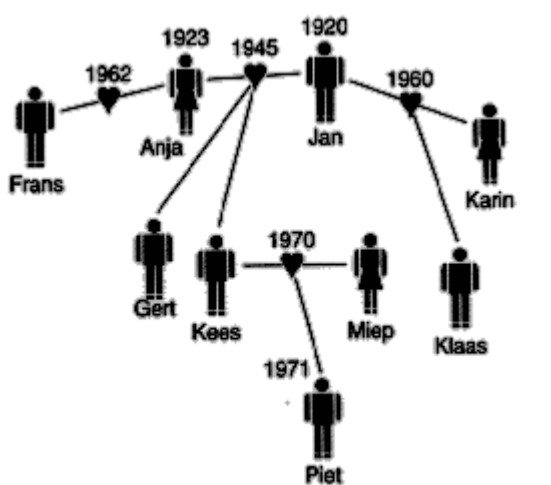
51 Verwoord deze voorbeelden, met andere woorden zoek zinsrepresentaties voor die feiten in natuurlijke taal.

Onderken ondeelbare (elementaire) feiten.

Inperking tot elementaire feiten komt voort uit de basisfilosofie van NIAM.

In termen van de ANSI/SPARC drie-lagen-architectuur voor informatiesystemen (extern, conceptueel en intern schema): het in kaart brengen van de elementaire feiten (de semantische atomen van informatie) garandeert een heldere afbakening tussen het centrale conceptueel schema (als FeitenGrammatica) en de externe schema's (voor diverse gebruikersapplicaties). Voor elk ondeelbaar feittype kan men immers éénvoudig aangeven of het wel of niet voorkomt in een gegeven extern schema. In NIAM kunnen externe schema's dan ook simpelweg als deelverzamelingen van het totale conceptuele schema worden gezien. Dit geldt niet voor de entiteitstypen en relatietypen uit het entity-relationship-model of voor de tabeltypen uit het relationele model. Voor de fijnproevers: NIAM beschouwt een relationeel database-schema van een gegevensbank dan ook als niets meer of minder dan het externe schema voor de gegevensbankbeheerder.

Aan de hand van een voorbeeld (naar een oorspronkelijk idee van Nijssen) bespreken we kort de structurele aspecten van NIAM. Figuur 1 geeft een aantal feiten weer betreffende een familiestamboom in een aan de gebruikers vertrouwde vorm. Laat de onderstaande verzameling zinnen een verwoording zijn van de in figuur 1 aanwezige elementaire feiten (om technische redenen laten we de beginhoofdletter van de zinnen weg).



Figuur 1: Een familiestamboom

“Jan is een man.” (Z1), “Anja is een vrouw.” (Z2), “Kees is een man.” (Z3), “Gert is een man.” (Z4), “Klaas is een man.” (Z5), “Karin is een vrouw.” (Z6), “Piet is een man.” (Z7), “Miep is een vrouw.” (Z8), “Frans is een man.” (Z9), “Piet is geboren in 1971.” (Z10), “Jan is geboren in 1920.” (Z11), “Anja is geboren in 1923.” (Z12), “Jan is gehuwd met Anja.” (Z13), “Jan is gehuwd met Karin.” (Z14), “Kees is gehuwd met Miep.” (Z15), “Frans is gehuwd met Anja.” (Z16), “Het huwelijk van Jan en Anja is voltrokken in 1945.” (Z17), “Het huwelijk van Jan en Karin is voltrokken in 1960.” (Z18), “Het huwelijk van Kees en Miep is voltrokken in 1970.” (Z19), “Het huwelijk van Frans en Anja is voltrokken in 1962.” (Z20), “Kees is geboren uit het huwelijk van Jan en Anja.” (Z21), “Gert is geboren uit het huwelijk van Jan en Anja.” (Z22), “Klaas is geboren uit het huwelijk van Jan en Karin.” (Z23), “Piet is geboren uit het huwelijk van Kees en Miep.” (Z24).

In plaats van zin Z10 had de gebruiker ook kunnen verwoorden: “1971 is het geboortjaar van Piet.” (Z10’, of: “In 1971 kwam Piet ter wereld.” (Z10’)). Desgevraagd zegt onze gebruiker echter dat Z10, Z10’ en Z10’’ alle drie hetzelfde betekenen, dus verwoordingen zijn van hetzelfde feit. We kunnen een feit dus zien als de verzameling van alle zinnen met eenzelfde betekenis (meer precieze definitie: als de equivalentieklasse van alle zinnen onder de equivalentierelatie betekent hetzelfde als). Dit inzicht is van groot belang, omdat gebruikers zich vaak van verschillende zinsformuleringen bedienen voor overeenkomstige feiten. Dit onderscheid tussen feiten en zinnen is, voorzover wij weten, in de literatuur over informatiemodellering niet eerder gemaakt en heeft belangrijke consequenties (zie paragraaf 2.2).

De voorbeeldzinnen Z10, Z11 en Z12 zijn alle drie van hetzelfde zinstype “<PERSOON> is geboren in <JAAR>”. In dit zinstype geven de vishaken < > open plekken (rollen) aan, waarin aanduidingen voor dingen (objecten) ingevuld moeten worden. Daarbij geldt een sterke typering: op de eerste open plek <PERSOON> moet een aanduiding voor een object van het objecttype PERSOON worden ingevuld en op de tweede open plek <JAAR> een aanduiding van een object van het objecttype JAAR.

De bovenstaande verzameling zinsrepresentaties van elementaire feiten kunnen we nu classificeren tot een zestal elementaire feittypen, die we in onderstaand overzicht een naam hebben gegeven: GEBOORTE, HUWELIJK enzovoorts. Bij elk feittype zijn tevens één of meer representatieve elementaire zinstypen gegeven.

Bij het analyseproces zijn objecttypen onderkend zoals PERSOON en JAAR. Natuurlijk kunnen we in de zinstypen de objecten zelf niet invullen, maar objecten worden vaak 1-op-1 gekoppeld aan objecten die men wel kan opschrijven, de lexicale objecten ofwel labels: namen, nummers, codes enzovoorts. Er zijn in ons voorbeeld twee niet-lexicale objecttypen (ook wel entiteitstypen of dingtypen genoemd) met bijbehorend lexicaal objecttype (ook wel: naamtype of labeltype) aanwezig:

Feittype:	HUWELIJK	(FT4)
Zintypen:	"<PERSOON> is gehuwd met <PERSOON>"	(ZT4)
	"<PERSOON> en <PERSOON> zijn gehuwd"	(ZT4)
Objecttype- formuleringen:	'het huwelijk van <PERSOON> en <PERSOON>'	(OTF4)
	'<PERSOON> en <PERSOON>'s huwelijk'	(OTF4)

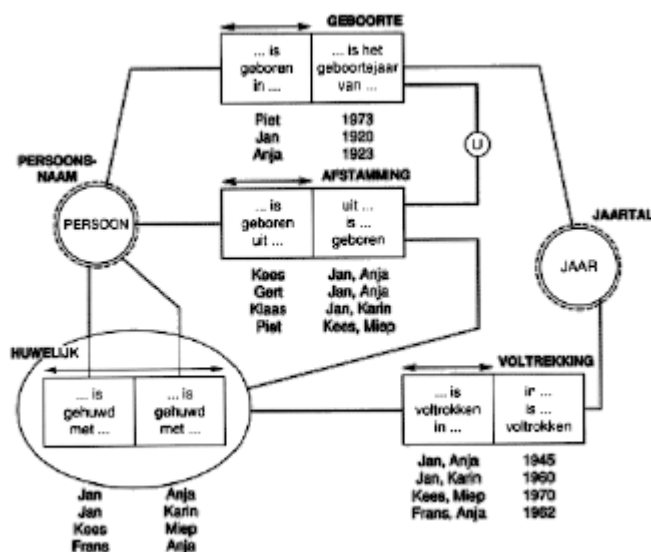
Naast de objecttypen PERSOON en JAAR speelt echter ook het objecttype HUWELIJK een rol in sommige feittypen, namelijk in FT5 en FT6. Maar HUWELIJK was ook een feittype: FT4. Sommige feittypen kunnen dus zelf optreden als objecttypen die rollen spelen in andere feittypen. Het op deze manier tot objecttype maken van een feittype wordt met een taalkundige termen ook wel nominalisatie genoemd (Nijssen, 1990, blz. 134).

Zoals het label 1945, als aanduiding voor een object van het type JAAR, in het zinstype ZT5 kan worden ingevuld op de invulplaats <JAAR>, kan op de invulplaats <HUWELIJK> de formulering ‘het huwelijk van Jan en Anja’ worden ingevuld als aanduiding voor een object van het type HUWELIJK. We noemen zo'n aanduiding als ‘het huwelijk van Jan en Anja’ een objectformulering van het feit dat wordt gerepresenteerd door de zin “Jan is gehuwd met Anja.” (Z13). Objectformuleringen van feiten zijn zelf geen volledige zinnen, maar zinsgedeelten, die op invulplaatsen kunnen worden ingevuld (taalkundigen zouden zeggen: naamwoordelijke groepen). Zo ontstaat dus de zin Z17 uit zinstype ZT5 door daarin deze objectformulering in te vullen op de

invulplaats <HUWELIJK> en het label 1945 op de invulplaats <JAAR>. Objectformuleringen van feiten gebruiken we dus net als labels, namelijk als aanduidingen voor objecten. Op typeniveau krijgen we dan, naast zinstypen als “<PERSOON> is gehuwd met <PERSOON>.” (ZT4) van het feittype FT4, ook objecttypeformuleringen daarvan, bijvoorbeeld ‘het huwelijk van <PERSOON> en <PERSOON>’ (OTF4). Zo krijgen we dus:

Objecttype:	Labeltype:	Voorbeelden van labels:
PERSOON	PERSOONS- NAAM	'Jan', 'Anja'
JAAR	JAARTAL	'1920', '1923'

Naar analogie gebruiken we in plaats van de termen zin en zinstype ook wel: feitformulering en feittypeformulering.



Figuur 2: Informatiediagram

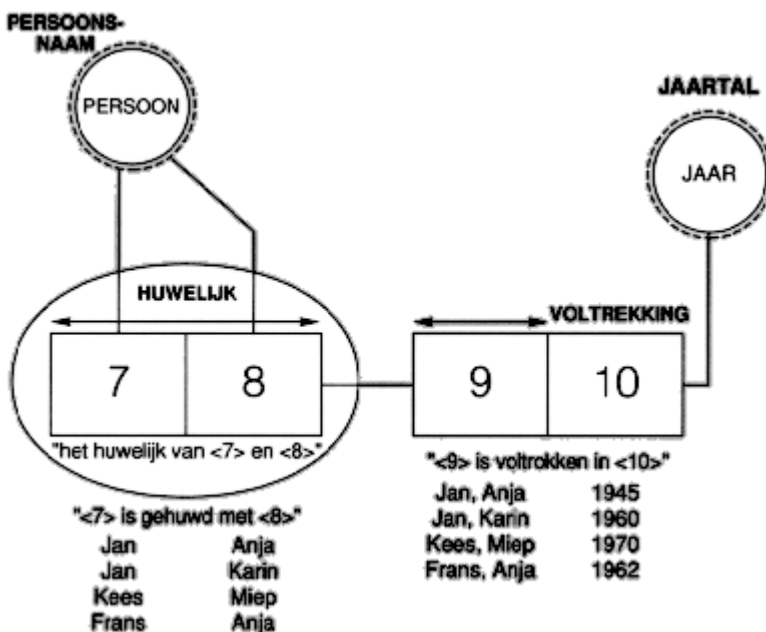
Het voorgaande is in beeld gebracht in het informatiediagram van figuur 2. Feittypen worden weergegeven als een rechthoek, die in evenveel deelrechthoeken is opgedeeld als er rollen zijn in de zinstypen, die het feittype representeren. Een 1-op-1-combinatie van een niet-lexicaal en een lexicaal objecttype wordt getekend als een doorgetrokken cirkel omgeven door een gestippelde cirkel (Wintraecken, 1985). (In andere literatuur: als alleen een getrokken cirkel waarin tussen haakjes de naam van het lexicale objecttype staat, zoals bij Nijssen & Halpin (1989) en Nijssen (1990).) Daarmee worden de deelrechthoeken verbonden, die de door de betreffende objecttypen gespeelde rollen representeren. In of naast de deelrechthoeken worden dan zinstypen van het betreffende feittype geschreven (met stippeltjes in plaats van de vishaken met de namen van de objecttypen), volgens de conventie dat de eerste rol van zo'n zinstype gespeeld wordt door het met die deelrechthoek verbonden objecttype. Een in genominaliseerde vorm gebruikt feittype, dat is een feittype dat ook als objecttype optreedt, wordt omgeven door een getrokken cirkel of ovaal, zodat de rollen die dit genominaliseerde feittype vervult, daaraan kunnen aangrijpen. Deze tekenwijze geeft de mogelijkheid het diagram te bevolken met de labels uit de zinnen, die de voorbeeldfeiten verwoorden. Uniciteitsbeperkingsregels (of uniciteitsregels) (Nijssen & Halpin, 1989, blz. 66 e.v.) worden aangegeven met tweepuntige pijlen boven de betreffende rollen, of, indien een uniciteitsregel rollen uit verschillende feittypen betreft (buitenom-uniciteitsregel), met een klein cirkeltje, waarin de letter U, met verbindingslijnen naar de betreffende rollen.

Opmerkingen:

In figuur 2 hebben we de feittypen FT1 en FT2 weggelaten, evenals de beperkingsregels (deelverzamelingsregels volgens Nijssen & Halpin (1989), blz. 177 e.v.) voor de rollen die het objecttype **PERSOON** speelt in het feittype **HUWELIJK**: namelijk dat de eerste respectievelijk de tweede rol slechts bevolkt mag worden met labels die ook voorkomen in de bevolking van het feittype **MAN** respectievelijk in die van het feittype **VROUW**. De reden is dat we ons in dit artikel niet willen belasten met een bespreking van het begrip subtype. In paragraaf 2.2 komen we hierop nog terug.

We zijn er impliciet van uitgegaan dat de feitenverzameling van figuur 1 significant is (dat wil zeggen dat de voorbeelden alle toegestane situaties tonen) voor ons UoD. Bijvoorbeeld: dezelfde twee personen kunnen slechts eenmaal met elkaar huwen. Daarom konden we verwoorden: "... is gehuwd met ...” zonder informatieverlies en konden we spreken van ‘hèt huwelijk van ... en ...’. Als dezelfde twee personen meer dan eens met elkaar zouden kunnen huwen, dan zouden we qua beperkingsregels een enigszins andere grammatica krijgen. Verder: de buitenom-uniciteitsregel in figuur 2 verbiedt het vóórkomen van tweelingen. Hoewel dit weinig realistisch is, vonden we het belangrijk een voorbeeld van zo'n buitenom-uniciteitsregel op te nemen.

Het schrijven van zinstypen in de deelrechthoeken die de rollen representeren, zoals in figuur 2 werd gedaan, suggereert ten onrechte een 1-op-1-correspondentie tussen de rollen in en de zinstypen voor een feittype. In Nijssen & Halpin (1989, blz. 298) en Nijssen (1990, blz. 77 e.v.) worden deze zinstypen op meta-grammaticaal niveau zelfs gebruikt als labels om de rollen aan te duiden (rolnamen). Ook in een aantal op NIAM gebaseerde CASE-tools is deze meta-grammaticale denkfout ingebouwd, die dan onder andere een blokkade vormt voor het genereren van zinsrepresentaties van feiten. We prefereren daarom een notatie als in figuur 3, waarbij de rollen een unieke code krijgen en onder ieder feittype tenminste één zinstype (feittypeformulering) wordt geschreven waarin deze codes terugkomen en bij feittypen die zelf weer rollen spelen tenminste ook één objecttypeformulering.



Figuur 3: Feittypen met zinstypen en objecttypeformuleringen

2.2 Equivalentie van feittypen en objecttypen

Elk feittype kan op de beschreven wijze worden genominaliseerd. Als voorbeeld hiervan nemen we het feittype **GEBOORTE** (FT3) met bijbehorend zinstype “<PERSOON> is geboren in <JAAR>.” (ZT3). Een objecttypeformulering van dit zinstype is: ‘de geboorte van <PERSOON> in <JAAR>’.

Opmerking: in de analysefase in n-air nested NIAM kan het genest gebruik van feittypen, die een rol bevatten zonder uniciteitsregel daarboven, niet voorkomen (zie ook paragraaf 4.3). Hier echter is de vaststelling, dat ieder feittype nominaliseerbaar is (dus ook te zien is als objecttype) van belang om de equivalentie van feittypen en objecttypen aan te tonen.

Omgekeerd zijn ook alle objecten te zien als feiten. Labels, zoals Anja en 1923, die we kunnen invullen op de invulplaatsen <PERSOON> en <JAAR> in zinstype ZT3 van het feittype GEBOORTE (FT3), kan men zien als objectformuleringen voor feiten die worden uitgedrukt in de zinnen er bestaat een persoon met de naam Anja. En er is een jaar van belang met jaartal 1923. Het wezenlijke punt hier is dit: door een gebruiker uitgesproken zinnen als “Anja is geboren in 1923.” (Z12) zijn voor een informatie-analist, behalve zinsformuleringen voor een feit, óók een trigger voor de (impliciete) zinnen “Er bestaat een persoon met de naam Anja.” (Z25) en “Er is een jaar van belang met jaartal 1923.” (Z26), juist omdat de informatie-analist (als informatie-analist) in het fysieke bestaan van deze persoon en dat jaar niet is geïnteresseerd, maar slechts in zinnen waarin ze rollen spelen. Dit aspect kunnen we op enigszins overdreven wijze doen uitkomen in het volgende alternatief voor zin Z25: “Kennelijk bestaat er in dit UoD een persoon met de naam Anja.”.

Als zinstypen krijgen we dan “Er bestaat een persoon met de naam <PERSOONSNAAM>” (ZT7) en “Er is een jaar van belang met jaartal <JAARTAL>.” (ZT8). Dit zijn zinstypen die unaire feittypen representeren: er is maar één invulplaats, die in dit geval een lexicaal objecttype betreft. We noemen deze feittypen PERSOON (FT7) en JAAR (FT8) en geven ook enkele objecttypeformuleringen voor deze feittypen:

‘de persoon met de naam <PERSOONSNAAM>’ (OTF7)

‘<PERSOONSNAAM>’ (OTF7)

‘het jaar <JAARTAL>’ (OTF8)

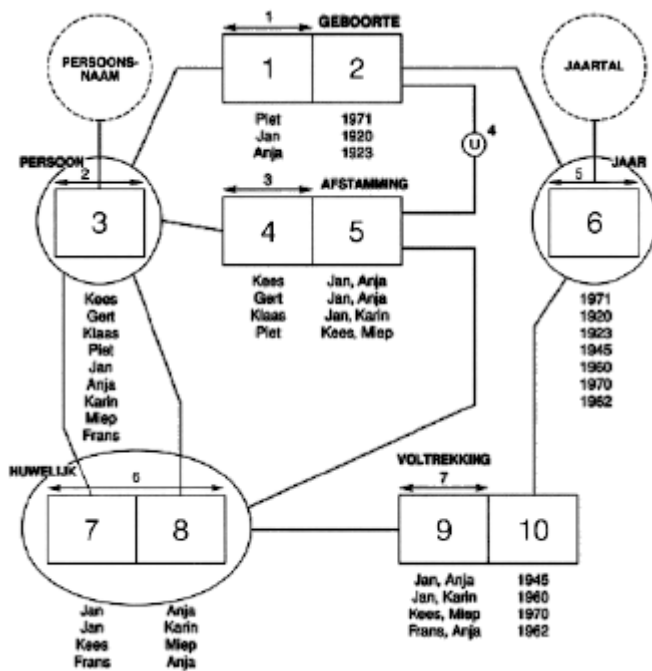
‘<JAARTAL>’ (OTF8)

De niet-lexicale objecttypen PERSOON en JAAR kunnen we dus beschouwen als nominalisaties van de feittypen PERSOON en JAAR betreffende de existentie (in het onderhavige UoD) van objecten die kunnen worden aangeduid met labels van het type PERSOONSNAAM en JAARTAL. In paragraaf 2.1 zagen we reeds dat ook het feittype HUWELIJK in genominaliseerde vorm rollen speelde in andere feittypen. Het is verhelderend dat ook nog even in dit licht te bezien: naast het zinstype “<PERSOON> is gehuwd met <PERSOON>.” (ZT4) hebben we (weer enigszins overdreven) ook “Er is in dit UoD een huwelijk van belang van <PERSOON> en <PERSOON>.”. De daaruit voortvloeiende objecttypeformulering ‘het huwelijk van <PERSOON> en <PERSOON>’ (OTF4) van het feittype HUWELIJK werd dan ook reeds gebruikt in de zinstypen ZT5, ZT5’, ZT6 en ZT6’.

We concluderen:

Alle feittypen zijn te nominaliseren tot objecttypen en alle (niet-lexicale) objecttypen zijn te zien als nominalisaties van feittypen.

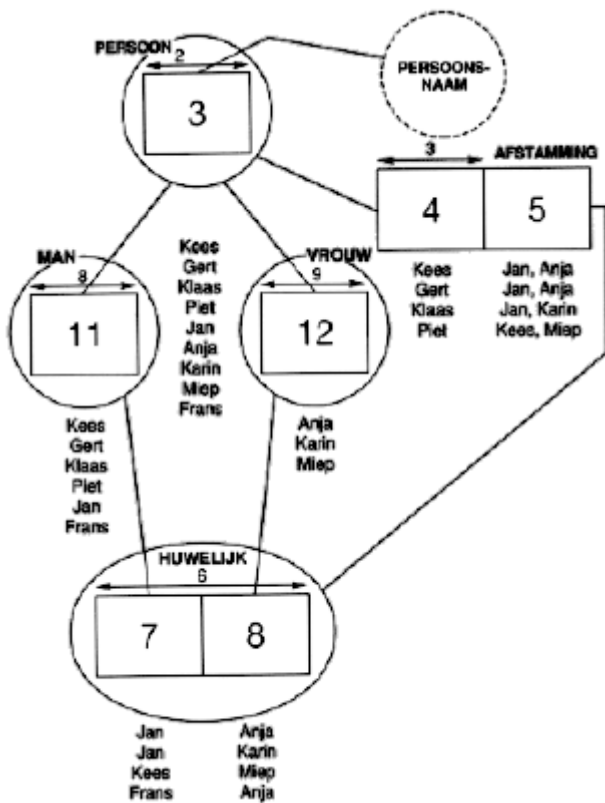
In figuur 4 staat nogmaals het informatiediagram van figuur 2, nu met de (niet-lexicale) objecttypen PERSOON en JAAR als nominalisaties van de unaire feittypen PERSOON en JAAR (let ook op de bevolkingen daarvan). We zouden desgewenst ook de feittypen GEBOORTE, AFSTAMMING en VOLTREKKING wel met een ovaal kunnen omgeven om te benadrukken dat ze ook als objecttypen kunnen worden gezien, maar dat is hier niet nodig omdat deze objecttypen geen rollen spelen in andere feittypen. Zinstypen en objecttypeformuleringen voor de feittypen hebben we uit plaatsgebrek weggelaten.



Figuur 4: Informatiediagram na unificatie feittype/objecttype

We unificeren nu de begrippen feittype en niet-lexicaal objecttype (entiteittype, dingtype) tot één begrip. Welk van beide termen we voor dit begrip gebruiken is dan eigenlijk niet zo belangrijk meer, hoewel de keuze voor feittype de communicatie-georiënteerde filosofie beter weerspiegelt. We zullen echter in paragraaf 3 ook wel de term objecttype gebruiken als generieke term voor niet-lexicaal objecttype (ofwel: feittype) en lexicaal objecttype (labeltype), waarbij we ons moeten realiseren dat een labeltype geen nominalisatie is van een feittype, maar slechts een verzameling (beter misschien: bron, source) van labels. Deze unificatie van niet-lexicale objecttypen en feittypen betekent een conceptuele versimpeling, want we hebben het aantal basisconcepten van NIAM met minstens één teruggebracht.

Ook subtypen zijn op eenvoudige wijze in deze visie onder te brengen, zodanig dat de drie begrippen feittype, objecttype en subtype uit NIAM genificeerd worden tot één begrip. Dit is met name ook van belang omdat het subtypebegrip (veelal onder de naam specialisatie en generalisatie) een belangrijke rol speelt in discussies over zogeheten object-oriented en semantische databases. We volstaan hier met de vaststelling dat de twee unaire feittypen MAN (FT1) en VROUW (FT2), gerepresenteerd door de zinstypen ZT1 en ZT2 of desgewenst door “Er bestaat een man gelijk aan <PERSOON>.” en “Er bestaat een vrouw gelijk aan <PERSOON>.”, in feite niets anders zijn dan existentie-postulerende feittypen/objecttypen, in dezelfde zin als de feittypen PERSOON en JAAR. Objecttypeformuleringen voor MAN en VROUW zijn dan bijvoorbeeld: ‘de man <PERSOON>’ en ‘de vrouw <PERSOON>’. Een zinstype voor het feittype HUWELIJK kan dan zijn: “<MAN> is gehuwd met <VROUW>.”. Zie figuur 5. Hierover is zeer veel meer te zeggen, bijvoorbeeld dat dergelijke feittypen vaak afleidbaar zijn uit andere feittypen. We laten deze hele problematiek in dit artikel echter rusten en keren terug tot het diagram van figuur 4. Wel merken we nog op dat deze visie op subtypen ook reeds voorkomt bij Wintraecken (1985, blz. 361), waar ze echter direct wordt ingeruild voor een meer objectgeoriënteerde kijk.



Figuur 5: Subtype-definiërende feittypen

Tenslotte: Hoe komt men nu van ingewikkelde nominalisaties in de grammatica weer tot de elementaire voorbeeldzinstypen en voorbeeldzinnen? Als voorbeeld nemen we het feittype FT6 met zinstype ZT6: “<PERSOON> is geboren uit <HUWELIJK>.” In eerste instantie vullen we op de invulplaats <HUWELIJK> de objecttypeformulering ‘het huwelijk van <PERSOON> en <PERSOON>’ in, zodat we het zinstype krijgen: “<PERSOON> is geboren uit het huwelijk van <PERSOON> en <PERSOON>.”. Vervolgens substitueren we voor de drie invulplaatsen <PERSOON> daarin de objecttypeformulering <PERSOONSNAAM> (OTF7) voor het feittype PERSOON, zodat we krijgen: “<PERSOONSNAAM> is geboren uit het huwelijk van <PERSOONSNAAM> en <PERSOONSNAAM>.” en tenslotte vullen we daarin de drie labels Kees, Jan en Anja in, zodat weer de voorbeeldzin Z21 ontstaat: “Kees is geboren uit het huwelijk van Jan en Anja.”. Dit proces van lexicaliseren (zie ook paragraaf 3.2) is eenvoudig te formaliseren en dus automatiseerbaar.

3 Van NIAM-grammatica tot ONF relationeel schema

In paragraaf 3.1 behandelen we de vastlegging van een NIAM-informatiegrammatica (zoals het diagram van figuur 6) in een relationele gegevensbank. In paragraaf 3.2 passen we de operaties groeperen en lexicaliseren toe op de in paragraaf 3.1 verkregen gegevensbank. In paragraaf 3.3 laten we zien dat nu een ONF relationeel database-schema is verkregen.

3.1 De vastlegging van een NIAM-informatiediagram

Beschouw de NIAM-grammatica van figuur 4, dat wil zeggen de figuur zonder de populatie. We zien een verzameling objecttypen: niet-lexicale (doorgetrokken cirkels of ovalen) en lexicale (gestippeld).

Verder zien we een verzameling rollen (de deelrechthoeken). Een rol zit altijd in precies één niet-lexicaal objecttype en is altijd de rol van (dat houdt in: wordt gespeeld door) precies één lexicaal of niet-lexicaal objecttype (deelrechthoek en ovaal verbonden door lijn). Uniciteitsregels staan boven rollen, waarbij in het algemeen een rol onder meer dan één uniciteitsregel kan vallen en een uniciteitsregel rollen uit meer dan één feittype kan betreffen. Deze NIAM-grammatica van figuur 4 kan nu volledig worden vastgelegd in een relationele gegevensbank (zie de vier tabellen van MG-populatie 1), waarvan het database-schema is (het teken ; betekent: optioneel, het tegengestelde van NOT NULL):

OBJECTTYPE [OTNAAM, SOORTCODE, DTNAAM_j, LENGTE_j]

(DTNAAM en LENGTE zijn dan en slechts dan relevant als SOORTCODE = 'L')

ROL [ROLNR, IN_OTNAAM, PREFIX_j, AAN_OTNAAM]

(de kolommen IN_OTNAAM en AAN_OTNAAM zijn verwijzende sleutels naar tabel OBJECTTYPE, waarbij voor het objecttype waarnaar IN_OTNAAM verwijst geldt, dat SOORTCODE = 'N')

UNICITEITSREGEL [URNR, ROLNR]

DEELVERZAMELINGSREGEL [DVRNR, VAN_ROLNR, NAAR_ROLNR]

(ook DVRNR, NAAR_ROLNR is een kandidaatsleutel)

Tabel OBJECTTYPE			
OTNAAM	SOORTCODE	DTNAAM	LENGTE
PERSOON	N		
PERSOONSNAAM	L	CHAR	30
JAAR	N		
JAARTAL	L	NUM	4
HUWELIJK	N		

Tabel ROL			
ROLNR	IN_OTNAAM	PREFIX	AAN_OTNAAM
2	PERSOON	GEB_	JAAR
3	PERSOON		PERSOONSNAAM
5	PERSOON		HUWELIJK
6	JAAR		JAARTAL
7	HUWELIJK	MANS	PERSOON
8	HUWELIJK	VROUWS_	PERSOON
10	HUWELIJK		JAAR

Tabel UNICITEITSREGEL		Tabel DEELVERZAMELINGSREGEL		
URNR	ROLNR	DVRNR	VAN_ROLNR	NAAR_ROLNR
2	3			
4	2			
4	5			
5	6			
6	7			
6	8			

MG-populatie 1

Dit relationele database-schema is een aanzet tot een metagrammatica (MG) voor onder andere NIAM-feitengrammatica's (FG's). We presenteren deze kern van een MG hier in een relationele vorm, omdat we denken dat deze het meest compact is. Deze MG verdraagt de NIAM-grammatica van figuur 4 als populatie, maar, zoals we zullen zien, ook andere dan NIAM-grammatica's. Het is dus geen (kern van een) NIAM-metagrammatica, maar een meer generieke metagrammatica.

Naar aanleiding hiervan een aantal opmerkingen:

We zouden deze generieke MG ook als NIAM-grammatica kunnen presenteren, maar dat laten we

hier rusten.

We hebben enkele kolommen toegevoegd, die in dit stadium niet wezenlijk zijn. De kolommen DTNAAM en LENGTE in de tabel OBJECTTYPE spelen straks een rol bij het genereren van een relationeel schema. De kolom PREFIX in de tabel ROL is van belang bij het automatisch genereren van relationele kolomnamen: indien twee rollen binnen één objecttype (feittype) gespeeld worden door hetzelfde objecttype (zoals de rollen 7 en 8 in ons voorbeeld), dan is het nodig om prefixen te geven. In andere gevallen kan het tot een helderder kolomnaam leiden (bijvoorbeeld de prefix van rol 2).

In de tabel DEELVERZAMELINGSREGELS kunnen deelverzamelingsregels worden vastgelegd. Hoewel deze in het informatiediagram van figuur 4 niet aanwezig zijn, zodat de tabel leeg blijft, willen we deze mogelijkheid toch in de MG opnemen, om redenen die in paragraaf 3.2 duidelijk zullen worden.

De gepresenteerde MG is niet volledig: de mogelijkheid voor het vastleggen van de feittype- en objectformuleringen (zie figuur 3) ontbreekt. Dat is onder andere van belang voor het automatisch genereren van zinsrepresentaties van feiten in natuurlijke taal, zoals aan het eind van paragraaf 2.2 is aangegeven. De formalisering hiervan is echter meer iets voor een vervolgartikel. Ook ontbreekt bijvoorbeeld de notie van totaliteitsbeperkingsregels (Nijssen & Halpin, 1989) en allerlei andere constraints.

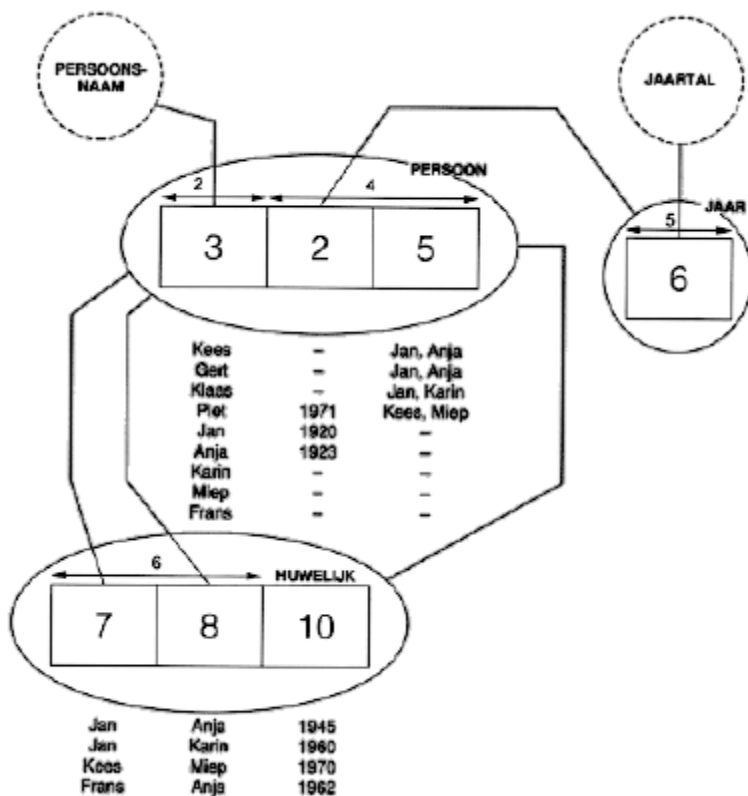
Hoe kunnen we verifiëren of hiermee de grammatica van figuur 4 volledig weergegeven is? De lezer kan dit nagaan door vanuit de gegeven tabellen weer een informatiediagram te tekenen. Dit zal dan inderdaad in essentie de in figuur 4 weergegeven grammatica zijn.

3.2 Groeperen en lexicaliseren

Na het vastleggen van het resultaat van een informatie-analyse volgt vaak de wens hieruit een relationeel database-schema af te leiden. We krijgen dan met niet-elementaire feit(typ)en te maken. Bij zo'n afleiding zijn de deelprocessen groeperen en lexicaliseren van belang, die we afzonderlijk beschouwen.

Groeperen

We geven eerst een nieuw informatiediagram van ons UoD, dat uit die van figuur 4 kan worden verkregen door het groeperen van feittypen. Hierbij worden alle binaire feittypen met één smalle uniciteitsregel samengenomen rond een gemeenschappelijk niet-lexicaal objecttype (zie Nijssen & Halpin, 1989, blz. 255).



Figuur 6: Gegroepeerd informatiediagram

Het diagram van figuur 6 wordt uit dat van figuur 4 verkregen door de feittypen PERSON, GEBOORTE en AFTAMMING samen te nemen tot één nieuw ternair, niet-elementair feittypen PERSON en ook de feittypen HUWELIJK en VOLTREKKING samen te trekken. Zo krijgen we:

Feittypen: PERSON (FT_{7i})

Zintypen: is geboren in uit (ZT_{7i})

Feittypen: HUWELIJK (FT_{4i})

Zintypen: en zijn gehuwd in (ZT_{4i})

Merk op dat bij de bevolking van deze nieuwe feittypen PERSON en HUWELIJK NULL-waarden nodig zijn. Dit is het gevolg van het opgeven van het elementair zijn van de feittypen: bij de bevolking van elementaire feittypen kan zich dat niet voordoen.

De bevolking van de generieke MetaGrammatica (MG) wordt nu gegeven in MG-populatie 2.

Tabel OBJECTTYPE			
OTNAAM	SOORTCODE	OTNAAM	LENGTE
PERSON	N		
PERSOONSNAAM	L	CHAR	30
JAAR	N		
JAARTAL	L	NUM	4
GEBORTE	N		
AFSTAMMING	N		
HUWELIJK	N		
VOLSTREKKING	N		

Tabel ROL			
ROLNR	IN_OTNAAM	PREFIX	AAN_OTNAAM
1	GEBORTE		PERSON
2	GEBORTE	GEB_	JAAR
3	PERSON		PERSOONSNAAM
4	AFSTAMMING		PERSON
5	AFSTAMMING		HUWELIJK
6	JAAR		JAARTAL
7	HUWELIJK	MANS	PERSON
8	HUWELIJK	VROUWS_	PERSON
9	VOLSTREKKING		HUWELIJK
10	VOLSTREKKING		JAAR

Tabel UNGITEITSREGEL		Tabel DEELVERZAMELINGSPREGEL		
URNR	ROLNR	DVRNR	VAN_ROLNR	NAAR_ROLNR
1	1			
2	3			
3	4			
4	2			
5	5			
6	7			
6	8			
7	9			

MG-populatie 2

Lexicaliseren

Lexicaliseren betekent (zie paragraaf 2.2) op typeniveau het substitueren van objecttypeformuleringen van feittypen in zinstypen van andere feittypen, waarmee net zolang wordt doorgegaan tot alle invulplaatsen lexicaal zijn geworden. Dit proces passen we nu toe op het feittype PERSON (FT7i), dat we eerder door groepering verkregen. Er waren in het zinstype ZT7i nog twee niet-lexicale invulplaatsen: <JAAR> en <HUWELIJK>.

We substitueren voor beide een objecttypeformulering, bijvoorbeeld OTF8 of de kortere OTF8 ϕ respectievelijk OTF4 en krijgen dan:

Zintype: "<PERSOONSNAAM> is geboren in <JAARTAL>
uit het huwelijk van <PERSON> en
<PERSON>" (ZT7ii)

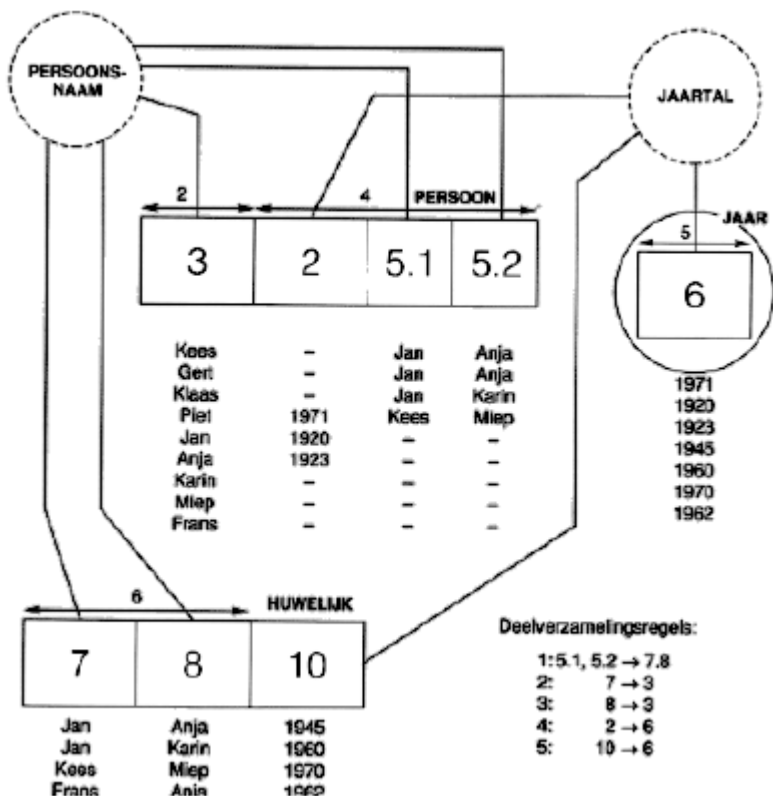
Nu zijn er nog steeds twee niet-lexicale invulplaatsen, beide van de vorm <PERSON>. Hiervoor substitueren we bijvoorbeeld OTF7 of het kortere OTF7 ϕ en verkrijgen zo:

Zintype: "<PERSOONSNAAM> is geboren in <JAARTAL>
uit het huwelijk van <PERSOONSNAAM> en
<PERSOONSNAAM>" (ZT7iii)

Voor het objecttype/feittype HUWELIJK verkrijgen we op dezelfde wijze:

Feittype: HUWELIJK (FT4iii)
Zintype: "<PERSOONSNAAM> is gehuwd met
<PERSOONSNAAM> in <JAARTAL>" (ZT4iii)

We kunnen dit lexicaliseringsproces ook grafisch uitvoeren op het informatiediagram van figuur 6 en krijgen dan figuur 7. Nu dienen ook deelverzamelingsregels (verwijzingen) gegenereerd te worden. Immers als het invullen van Piet,1971,Kees en Miep in bijvoorbeeld ZT7iii een bestaand feit verwoordt, dan moet ook gelden er is een jaar 1971 van belang. en Kees is gehuwd met Miep.,dus moet de bevolking onder de rol met rolnummer 2 een deelverzameling zijn van die onder rol 6 en de bevolking onder de rolcombinatie 5.1, 5.2 moet ook voorkomen onder de rolcombinatie 7, 8. Evenzo moeten de bevolkingen onder de afzonderlijke rollen 7 en 8 gezamenlijk voorkomen onder rol 3.



Figuur 7: Gegroepeerd en gelexicaliseerd informatiediagram

In MG-populatie 3 is de nieuwe inhoud van de MG te lezen.

Tabel OBJECTTYPE			
OTNAAM	SOORTCODE	DTNAAM	LENGTE
PERSOON	N		
PERSOONSNAAM	L	CHAR	30
JAAR	N		
JAARTAL	L	NUM	4
HUWELIJK	N		

Tabel ROL			
ROLNR	IN_OTNAAM	PREFIX	AAN_OTNAAM
2	PERSOON	GEB_	JAARTAL
3	PERSOON		PERSOONSNAAM
5.1	PERSOON	MANS_	PERSOONSNAAM
5.2	PERSOON	VROUWS_	PERSOONSNAAM
6	JAAR		JAARTAL
7	HUWELIJK	MANS_	PERSOONSNAAM
8	HUWELIJK	VROUWS_	PERSOONSNAAM
10	HUWELIJK		JAARTAL

Tabel UNICITEITSREGEL		Tabel DEELVERZAMELINGSREGEL		
URNR	ROLNR	DVRNR	VAN_ROLNR	NAAR_ROLNR
2	3	1	5.1	7
4	2	1	5.2	8
4	5.1	2	7	9
4	5.2	3	8	3
5	6	4	2	6
6	7	5	10	6
6	8			

MG-populatie 3

3.3 Relatieve schema

Om te komen tot een relationeel database-schema passen we onderstaande (bijjectieve) transformatie toe op onze terminologie na het groeperen en lexicaliseren. Dan blijkt dat we in de laatste bevolking van onze generieke metagrammatica (MG-populatie 3) en dus ook in figuur 7 een relationeel schema hebben gekregen.

feittype/objecttype	↔	tabel(schema) of relatieschema
rol	↔	kolom of attribuut
lexicaal objecttype	↔	domein
uniciteitsregel	↔	sleutel
deelverzamelingsregel	↔	verwijzing

Met behulp van een rapportgenerator kunnen we dan ook met deze laatste populatie als input voor een concreet RDBMS het onderstaande Rapport 1 genereren. (N.B.: NOT NULL-kwalificaties laten we buiten beschouwing, omdat we ook totaliteitsregels niet beschouwen; zie paragraaf 3.1, opmerking 4.) Ook definities voor kandidaatsleutels, verwijzingen en dergelijke kunnen zo worden gegenereerd (indien deze door het RDBMS worden ondersteund). De tabel JAAR zouden we waarschijnlijk schrappen (de betrekkelijk eenvoudige regels voor het schrappen van sommige tabellen laten we hier rusten). Dan is het verkregen database-schema zelfs in optimale normaalvorm (ONF) (Leung & Nijssen, 1987). Overigens leidt uitsluitend lexicaliseren van een n-air nested NIAM-grammatica ook tot een relationeel schema dat reeds in 5NF zal zijn, maar in het algemeen niet in ONF.

```

create table PERSOON (
  , PERSOONSNAAM          CHAR (30)
  , GEB_JAARTAL           NUM  (04)
  , MANS_PERSOONSNAAM    CHAR (30)
  , VROUWS_PERSOONSNAAM CHAR (30)
);

create table JAAR (
  JAARTAL           NUM  (04)
);

create table HUWELIJK (
  MANS_PERSOONSNAAM CHAR (30)
  , VROUWS_PERSOONSNAAM CHAR (30)
  , JAARTAL           NUM  (04)
);

```

Rapport 1

We concluderen dat het op de in paragraaf 3.2 beschreven wijze afleiden van een relationeel schema uit een NIAM-feitengrammatica (FG) eenvoudig kan gebeuren middels gewone database-operaties op de populatie van de tabellen van de generieke metagrammatica (MG), nadat die in een database geïmplementeerd zijn.

4 Informatiemodel-specifieke metagrammatica's

In deze paragraaf willen we laten zien hoe een aantal bekende informatiemodellen kunnen worden gezien als speciale gevallen van het gepresenteerde generieke informatiemodel, door daaraan modelspecifieke beperkingsregels toe te voegen. Dit zal niet steeds in extenso worden aangetoond, deels omdat daarvoor hier de ruimte ontbreekt, deels omdat dit niet altijd exact mogelijk is gezien de weinig formele wijze waarop de informatiemodellen soms worden gedefinieerd.

4.1 Het relationele model

In paragraaf 3 zagen we reeds dat een volledige lexicalisering van een n-air nested NIAM-grammatica leidt tot een relationeel schema. Omgekeerd geldt dat onze generieke MG ieder relationeel schema als populatie kan bevatten. Zo'n populatie is dan vanzelfsprekend volledig gelexicaliseerd, omdat alle relationele attributen op (lexicale) domeinen zijn gedefinieerd. Dit betekent dat de gepresenteerde generieke MG specifiek wordt voor het relationele model door (naast een terminologietransformatie; zie paragraaf 3.3) slechts de volgende beperkingsregel op te leggen:

Alle rollen worden gespeeld door lexicale objecttypen.

Via een formele mathematische beschrijving van het relationele model (Maier, 1983; Remmen, 1982) en een mathematische beschrijving van ons generieke model, is het onzes inziens dan in principe mogelijk bovenstaande bewering formeel te bewijzen. Dit valt echter buiten de opzet van dit artikel.

4.2 Het semantische model

We definiëren een toekenning (attribuuttype of functie) als een binair feittype met minstens één rol die door een niet-lexicaal objecttype wordt gespeeld en waarvoor er een uniciteitsregel is, die uitsluitend die rol betreft. Iets slordiger: toekenningen zijn 1-op-n- of 1-op-1-feittypen, met tenminste aan de 1-kant een niet-lexicaal objecttype. In het informatiediagram van figuur 4 zijn GEBOORTE, AFSTAMMING en VOLTREKKING dus toekenningen.

We beschouwen nu het semantische model, zoals beschreven door Ter Bekke (Ter Bekke, 1988). Hierin wordt een entiteittype gezien als een zogenaamde aggregatie van de attributen van dat entiteittype. In het licht van ons grondmodel kunnen we een grammatica (schema) nu als

semantisch karakteriseren als het schema in zekere zin maximaal gegroepeerd is. Hiermee bedoelen we dat het voldoet aan de volgende beperkingsregel:

Er komen geen toekenningen voor.

In de grammatica van figuur 6 en de vastlegging daarvan in de MG (MG-populatie 2) komen geen toekenningen meer voor: ze zijn weggegrepeerd. Deze grammatica is dus te beschouwen als een semantisch schema (in de zin van Ter Bekke), dat we ook met de typerende syntax voor het semantische model kunnen weergeven (let op het essentiële gebruik van prefixen):

base PERSOONSNAAM (A30)

base JAARTAL (I4)

type JAAR = JAARTAL

type PERSOON = PERSOONSNAAM, HUWELIJK, GEBJAAR

type HUWELIJK = MANS_PERSOON, VROUWS_PERSOON, JAAR

De terminologietransformatie hierbij is:

lexicaal objecttype	↔	base (basistype)
feittytype/objecttype	↔	type
rol	↔	rolattribuut

Merkwaardig aan dit semantische model is dat we unieke identificatoren opgedrongen krijgen, ook als dat niet nodig is. Ter Bekke eist namelijk voor alle typen, dus ook voor PERSOON en HUWELIJK, bovendien nog een zogeheten externe representatie, te geven in termen van basistypen.

Samenvattend kunnen we de structuurcomponent van dit semantische model van Ter Bekke als volgt kenschetsen: wel volledig gegroepeerd, maar nog niet gelexicaliseerd.

Teneinde de kracht van onze generieke blik verder te onderstrepen nog een interessante vaststelling. In de nieuwe standaard voor SQL: SQL2 (SQL2, 1990) is het relationele domeinbegrip opgenomen. Beschouw nu de pseudo-SQL-statements in Rapport 2 en vergelijk ze met bovenstaande typedeclaraties van het semantische model. Dit is equivalent met de inhoud van de tabellen Ot en Rol in MG-populatie 2 en dus ook met bovenstaand semantisch schema. Helaas zijn bovenstaande SQL-statements volgens de SQL2-standaard echter niet correct, want daar mogen de kolomtyperingen alleen maar domeinnamen of concrete datatypen zijn (in onze terminologie: alleen namen van lexicaal objecttypen). We concluderen: wanneer in SQL2 als kolomtyperingen ook tabelnamen (dat wil zeggen: namen van niet-lexicale objecttypen) zouden zijn toegestaan, dan zou het mogelijk zijn alle tot nu toe besproken grammatica's in SQL2 weer te geven en in het bijzonder geldt dat dan ook voor een semantisch schema volgens Ter Bekke. De suggestie om SQL op deze manier uit te breiden gaf overigens ook Van der Lans al eens (Van der Lans, 1990, blz. 5.12).

```
create domain PERSOONSNAAM CHAR (30);
create domain JAARTAL NUM (04);

create table PERSOON (
  , PERSOONSNAAM PERSOONSNAAM
  , GEB_JAAR JAAR
  , HUWELIJK HUWELIJK
);

create table JAAR (
  JAARTAL JAARTAL
);

create table HUWELIJK (
  MANS_PERSOON PERSOON
  , VROUWS_PERSOON PERSOON
  , JAAR JAAR
);
```

Rapport 2

4.3 Het n-air nested NIAM-model

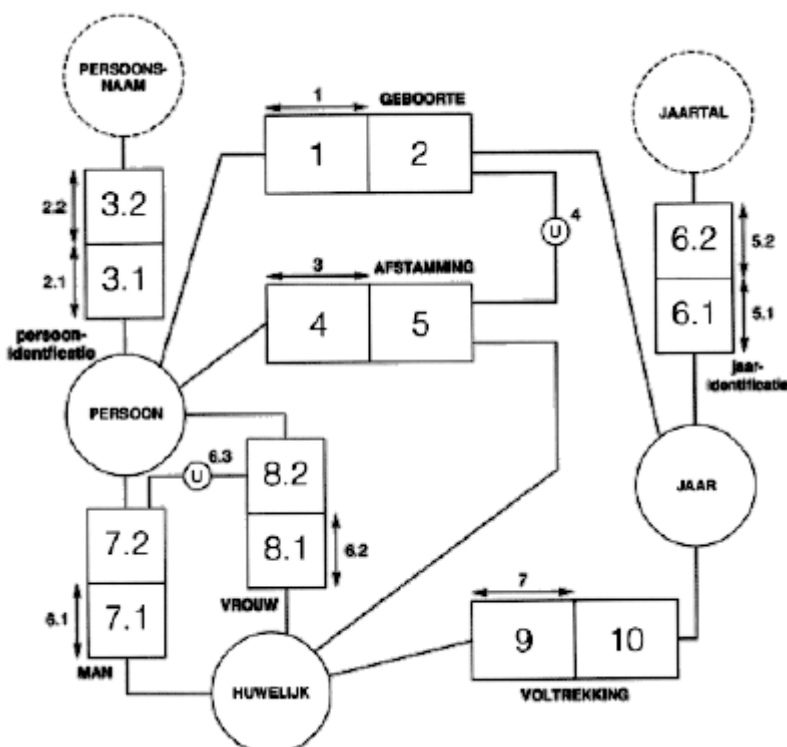
Het in paragraaf 3.1 gepresenteerde meta-database-schema ontstond vanuit een analyse van n-air nested NIAM. De beperkingsregel, die we er voornamelijk aan toe moeten voegen om een n-air nested NIAM-specifieke MG te krijgen, is de eis, dat alle feittypen elementair zijn (de informele definitie voor elementaire feittypen luidt: een feittype heet elementair als het niet is op te splitsen in kleinere feittypen zonder informatieverlies). Bij Nijssen & Halpin (1989, blz. 94 en paragraaf 5.3) en bij Nijssen (1990) zijn harde formele criteria voor het elementair-zijn van feittypen te vinden. Daarnaast hanteert Nijssen in Nijssen & Halpin (1989, blz. 87) voor het genominaliseerd gebruik van feittypen in andere feittypen (nesting) de eis dat in dat geval over alle n rollen een uniciteitsregel moet liggen (we zouden dat de n-regel voor nesting kunnen noemen). We concluderen dat voor een grammatica volgens het n-air nested NIAM-model (in de zin van Nijssen & Halpin, 1989) tenminste geldt:

Alle feittypen zijn elementair en nesting is slechts toegestaan voor feittypen waarvoor de n-regel geldt.

Opmerking. In wezen doet het er nu niet zoveel meer toe hoe we een n-air nested NIAM-grammatica in diagramvorm weergeven. We mogen een n-air nested grammatica desgewenst ook wel tekenen zoals in figuur 2 werd gedaan, mits we dit diagram begrijpen als een versuikering (syntactic sugar) voor het diagram van figuur 4, aangevuld met zinstypen zoals in figuur 3. Dit is een kleine structurele correctie op het n-air nested NIAM zoals beschreven bij Nijssen & Halpin (1989).

4.4 Het binaire model

Een manier om het elementair-zijn van feittypen te garanderen is te eisen dat alle feittypen binair of unair zijn.



Figuur 8: Een informatiediagram in binair NIAM

In figuur 8 is figuur 4 nogmaals getekend, waarbij echter het objecttype/feittypen HUIJELIJK is gedegroepeerd en de 1-op-1-correspondenties tussen de niet-lexicale objecttypen (NOLOT's) PERSOON en JAAR en de lexicale objecttypen (LOT's) PERSOONSNAAM en JAARTAL ook als binaire feiten zijn weergegeven. Dit diagram is typerend voor Wintraeckens binaire NIAM, dat ook geen nominalisatie toestaat (Wintraecken, 1985). Weliswaar versuikert Wintraecken de 1-op-1-feittypen tussen NOLOT's en LOT's (hij noemt ze brugtypen) op de wijze zoals in figuur 2 ook werd gedaan, maar (omdat hij geen nominalisatie toestaat en dus objecttypen niet kunnen worden gezien als nominalisaties van feittypen) moeten we deze brugtypen er wel achter denken. Dit brengt een fundamenteel probleem met zich mee: hoe moeten we ons de bevolking voorstellen onder rollen die door NOLOT's worden gespeeld? Welbeschouwd is er geen andere conclusie mogelijk dan: met pseudo-objecten. Dit is een ernstige structurele en methodologische tekortkoming.

Kunnen we dit model ook inpassen in ons generieke model? Ja, mits we de NOLOT's opvatten als feittypen met 0 (nul) rollen. Het binaire model kan in het licht van ons generieke model dus als volgt worden gekarakteriseerd:

Alle feittypen bevatten 0, 1 of 2 rollen en spelen zelf geen rollen.

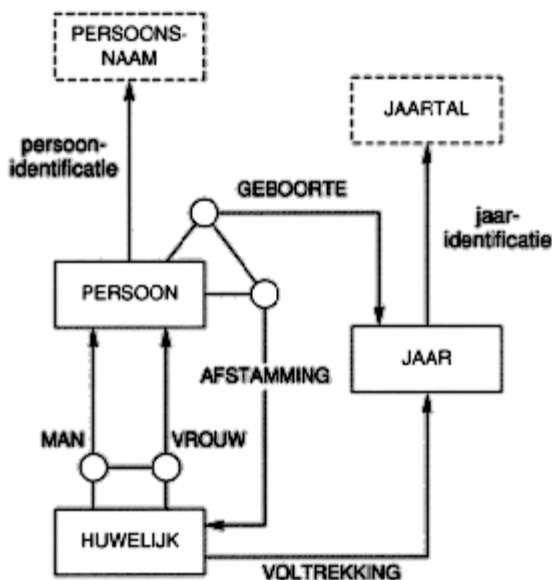
Andere auteurs (Shoval & Even-Chaime, 1987) eisen wel binariteit, maar staan nominalisatie van unaire feittypen en binaire feittypen met één uniciteitsregel over beide rollen wel toe. De discussie over de precieze vorm van de n-regel is dan niet nodig en de bevolking is mogelijk, mits we weer alle niet-lexicale objecttypen zien als nominalisaties van feittypen. Omdat in figuur 4 geen n-aire feittypen met $n = 2$ voorkomen, kunnen we dit diagram dus ook zien als een nested binair informatiediagram volgens Shoval, dat we desgewenst weer versuikerd mogen tekenen als in figuur 2.

4.5 Het functionele model

In het binaire model is het toegestaan om binaire veel-op-veel-relaties (feittypen met een uniciteitsregel over beide rollen) te hebben. Wanneer we ook die uitsluiten en alleen toekenningen (functies, zie paragraaf 4.2) toestaan, krijgen we een extreem binair model, dat feitelijk overeenkomt met het functionele model (Aerts & Van Hee, 1989). De generieke MG wordt dus specifiek voor dit model door het opleggen van de volgende beperkingsregel:

Alle feittypen bevatten 0 of 2 rollen. Alle binaire feittypen zijn toekenningen en spelen zelf geen rollen.

We volstaan hier met de opmerking dat door het opleggen van bovenstaande beperkingsregel een grammatica te zien is als een gerichte graaf (Bondy & Murty, 1976). Figuur 8 gaf een binair schema dat (toevallig) reeds extreem binair was, zodat het ook opgevat kon worden als een functioneel schema. In figuur 9 is het hertekend als graaf. Ook hier geldt echter dat zo'n schema niet bevolkbaar is, zodat we ook tegen het functionele model ernstige structurele en methodologische bezwaren hebben.



Figuur 9: Het informatiediagram als functionele graaf

Dat we met het opleggen van deze beperkingsregel een model krijgen dat equivalent is met het functionele model, kan in principe bewezen worden, aangezien er een exacte definitie van dit laatste model voorhanden is.

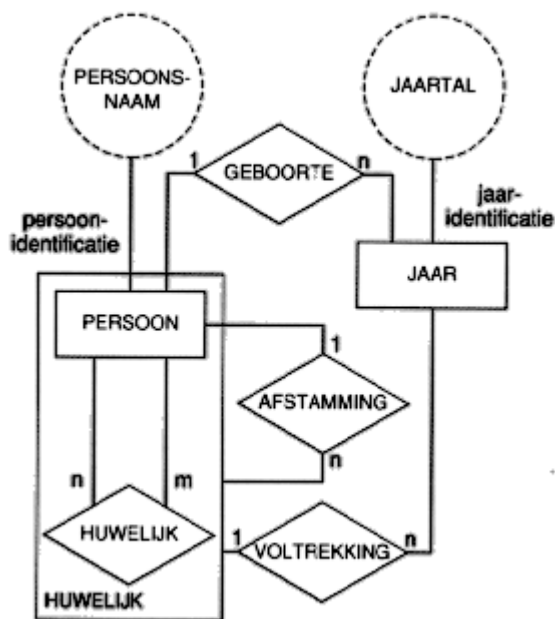
Opmerking: Omdat in het functionele model alle feittypen toekenningen zijn, is het te beschouwen als een tegenpool van het semantische model van Ter Bekke, dat immers geen toekenningen toestaat (zie paragraaf 4.2).

4.6 Het entity-relationship-model

Er zijn vele varianten van het entity-relationship-model (ER-model) (Elmasri & Navathe, 1989) in gebruik. In het zogeheten extended ER-model bestaat zelfs de mogelijkheid om relatietypen tot entiteitstypen te maken (te aggregeren tot entiteitstypen (Smith & Smith, 1977)) die weer kunnen participeren in andere relatietypen. Dit is vergelijkbaar met het nominaliseren en nesten van feittypen.

De belangrijkste vaststelling is dat onze feittypen/entiteitstypen in het ER-model worden opgedeeld in drie constructen: het entiteitstype, het relatietype en het attribuuttype. ER geeft echter nauwelijks voorschriften voor het gebruik van deze constructen (methodologisch zwakke onderbouwing) in het analyseproces, waardoor verschillende analisten gauw verschillende ER-diagrammen zullen tekenen.

Als voorbeeld geven we in figuur 10 een extended ER-diagram dat min of meer equivalent is met figuur 4 (zonder populatie), mits we daarin uniciteitsregel 4 weglaten. Door omgekeerd in een willekeurige ER-grammatica alle entiteitstypen, relatietypen en attribuuttypen te zien als feittypen (de eerste weer met 0 rollen) en functionaliteiten te zien als uniciteitsregels, kunnen we dat ER-diagram als populatie in onze generieke MG onderbrengen. Het generieke model omvat dus ook de verschillende varianten van het ER-model, maar een preciezere positionering is door de methodologische vaagheid van het ER-model niet goed mogelijk.



Figuur 10: Extended ER-diagram

5 Besluit

We hebben laten zien, dat consequente toepassing van het feittype-georiënteerde denken (communicatie-georiënteerde filosofie) leidt tot een bijzonder simpel en elegant generiek informatiemodel, waarvan de diverse informatiemodellen bijzondere gevallen blijken te zijn. We hebben de kern van een relationele versie van een metagrammatica van dit model besproken in paragraaf 3. In paragraaf 4 lieten we zien hoe andere gegevens/informatiemodellen vanuit dit perspectief begrepen kunnen worden. Deze resultaten zijn van belang voor de praktijk, de theorie en het onderwijs. We gaan daarop nog even kort in.

Veel van de ideeën in dit artikel zijn in de praktijk ontstaan. Bij de Nederlandse Spoorwegen wordt n-air nested NIAM met succes toegepast voor informatiemodellering van (bij de spoorwegen veel voorkomende) grafische informatiebronnen: emplacementstekeningen, tijdwegdiagrammen en dergelijke. Er ontstond daarbij behoefte aan een hulpmiddel om NIAM-grammatica's gestructureerd vast te leggen, onder andere voor het genereren van documentatie. Op basis van de in dit artikel geschetste architectuur heeft Van der Lek in zeer korte tijd een registratiehulpmiddel ontwikkeld (NiReG: NIAM Registratie en Groeperingstool) op basis van een simpel data-managementsysteem (Paradox). De groeperings- en lexicaliseringsalgoritmen (paragraaf 3.2) bleken daarmee ook eenvoudig te implementeren, zodat de benodigde datadefinitie-statements voor het gebruikte RDBMS (Rdb van DEC) konden worden verwezenlijkt met de rapportgenerator van Paradox. Aan de voorkant werden interfaces gemaakt met SDW-NIAM, een tekentool voor binair NIAM (n-air nested NIAM kon daarin geïmplementeerd worden via transformatie-afspraken) en in een later stadium met een speciaal voor n-air nested NIAM aangepaste versie van Excelerator.

De generalisatie van een aantal bekende informatiemodellen tot een generiek informatiemodel heeft vanuit theoretisch oogpunt het voordeel van wetenschappelijke economie: verdere theorievorming hoeft nu in het algemeen slechts één keer te gebeuren, namelijk voor het generieke model, en is dan ook van toepassing op de specifieke informatiemodellen. Voorbeelden van onderwerpen voor verdere theorievorming zijn:

- opnemen van verdere beperkingsregels (zoals totaliteitsregels) in de MG;
- vastlegging van zinstypen en objecttypeformuleringen in de MG en het genereren van zinsrepresentaties van populaties van databases in natuurlijke taal;

betere taalkundige verankering van het begrip elementair feittype (n-1 -regel) en genest gebruik van feittypen (n-regel);
theorie over semantisch equivalente grammatica's;
formele beschrijvingen van de diverse informatiemodellen en de beperkingsregels op de generieke MG die deze transformeren in modelspecifieke MG's.

Voor het onderwijs is de reductie van het aantal primitieve begrippen van belang. Waar voorheen een begrip als attribuut(type) in feite een ongedefinieerd basisbegrip was, kan men het nu zien als een speciaal soort feittype: de toekenning (paragraaf 4.2). Ook is het, gezien de veelheid van informatiemodellen waarmee informaticastudenten in aanraking komen, van belang de diverse modellen begripsmatig te kunnen plaatsen. Deze modellen stuk voor stuk min of meer uitputtend te behandelen is niet de juiste didactische weg, een typering vanuit een generiek model blijkt dan veel tijdswinst op te leveren en bovendien meer inzicht te geven. Uit zes jaar ervaring met onderwijs in informatiemodellering aan de Hogeschool Gelderland is gebleken dat het consequent hanteren van een communicatie-georiënteerde denkwijze aan studenten de helikopterblik kan geven, die het hun in stagesituaties mogelijk maakt zich flexibel aan te passen.

Dankwoord

Wij bedanken ons medelid van de NIAM-groep Kriss W. Yunker voor zijn constructieve opmerkingen en discussies.

Literatuur

- Aerts, A.T.M. en K.M. van Hee (1989), Modelleren met een functioneel datamodel, Informatie jaargang 31, nr. 12, blz. 901-1016.
- Bekke, J.H. ter (1988), Database ontwerp, Stenfert Kroese, Leiden, 2e druk.
- Bondy, J.A. en U.S.R. Murty (1976), Graph theory with applications, The Macmillan Press.
- Elmasri, R.A. en S.B. Navathe (1989), Fundamentals of database systems, Addison-Wesley, Reading (Mass.).
- Lans, R.F. van der (1990), Resultaten studie 15 SQL-produkten en nieuwe ontwikkelingen SQL, NOVI-studiedag 3 mei 1990.
- Leung, C.M.R. en G.M. Nijssen (1987), From a NIAM conceptual schema into the optimal SQL relational database schema, Collected papers of the 10th Australian Computer Science Conference.
- Maier, David (1983), The theory of relational databases, Computer Science Press.
- Nijssen, G.M. (1977), Integratie, informatiesystemen en informatiebasis, Informatie jaargang 19, nr. 4, blz. 188-201.
- Nijssen, G.M. en T.A. Halpin (1989), Conceptual schema and relational database design, Prentice-Hall, Englewood Cliffs (New Jersey).
- Nijssen, G.M. (1990), Grondslagen van bestuurlijke informatiesystemen, Nijssen Adviesbureau voor Informatica B.V.
- Remmen, F. (1982) Databases, grondslagen voor de logische structuur, Academic Service, Schoonhoven.
- Shoval, P. en M. Even-Chaime (1987), ADDS: A system for automatic database schema design based on the Binary-Relationship Model, Data & Knowledge Engineering 2, Elsevier Science Publishers, blz. 123-144.
- SQL2 (1990), Committee Draft Database Language SQL2, ISO 9075: 199x.
- Smith, J.M. en D.C.P. Smith (1977), Database abstractions: aggregation and generalisation, ACM Transactions on Database Systems, Vol. 2, Nr. 2, blz. 105-133.
- Wintraecken, J.J.V.R. (1985), Informatie-analyse volgens NIAM in theorie en praktijk, Academic Service, Schoonhoven.

Dr. Harm van der Lek is werkzaam bij BSO/Management Support als senior adviseur en is gespecialiseerd in gegevensanalyse en databases. Hij introduceerde het moderne (niet binaire) NIAM bij de Nederlandse Spoorwegen als methodologische standaard voor gegevensanalyse. Hij is voorzitter van de NIAM groep, een groep van personen uit bedrijfsleven en onderwijs die onderzoek doet met betrekking tot de grondslagen van feitgeoriënteerde informatie-, proces- en impulsmodellering.

Drs. Guido Bakema en drs. Jan Pieter Zwart zijn beide als docent informatica verbonden aan de Hogeschool Gelderland. Zij ontwikkelen en verzorgen o.a. cursussen over relationele database-systemen en informatiesysteemontwikkeling, met name aan de opleiding Docent informatica en de speerpunt HBO-opleiding Systeemontwikkeling, waarvoor n-air nested NIAM als methodologisch referentiekader is gekozen. Beide coauteurs zijn lid van de NIAM groep.