



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Het toepassen van modelleringstechnieken als Dataflow-
diagrammen en Entiteit-Relatie-diagrammen

S. Brinkkemper, L.J.M. Gerritsen, J.E.A. van Hintum
Afdeling Informatiesystemen
Universiteit van Nijmegen
Toernooiveld
6525 ED Nijmegen

Samenvatting

Modelleringstechnieken in de systeemontwikkeling kunnen effectiever en efficiënter worden toegepast middels modelleringsprocedures. Uitgaande van de concepten van een modelleringstechniek wordt een modelleringsprocedure ontwikkeld en geformaliseerd door middel van meta-modellering. Ter illustratie wordt een modelleringsprocedure voor Entiteit-Relatie-diagrammen besproken.

1 Inleiding

Voor het ontwerpen van informatiesystemen zijn een groot aantal verschillende soorten modelleringstechnieken in gebruik. Entiteit-Relatie-diagrammen voor gegevensmodellering en Dataflow-diagrammen voor procesmodellering zijn twee van de meer populaire technieken. Tegenwoordig ondersteunen moderne ontwerphulpmiddelen, ook wel Computer Aided Software Engineering tools (CASE-tools) genoemd, de ontwerper bij de toepassing van deze technieken in vaak complexe modelleringstaken.

Het gebruik van een modelleringstechniek in de praktijk blijkt echter veel intuïtie te vragen en gebaseerd te zijn op ervaring van vage, niet geformaliseerde heuristieken. Vaak blijkt ook dat allerlei betrokkenen in ontwikkelprojecten van informatiesystemen, zoals de opdrachtgevers en de toekomstige gebruikers, meestal niet op de hoogte zijn met de modelleringstechnieken en problemen hebben met notaties en concepten. Hierdoor zijn discussies over de correctheid van diagrammen heel moeilijk te voeren.

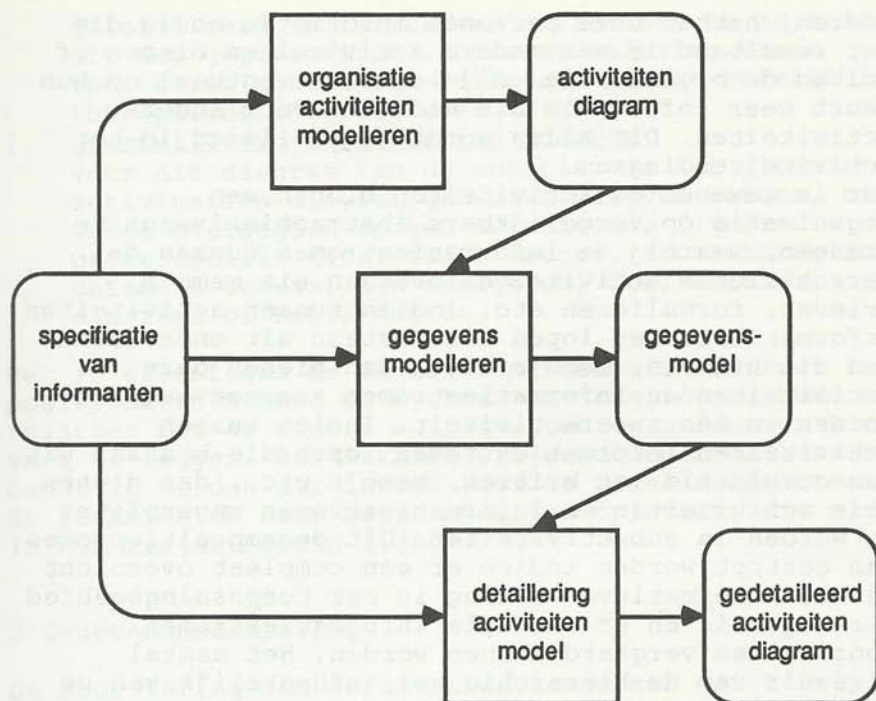
Ten einde deze situatie te verbeteren stellen we in dit artikel voor om voor modelleringstechnieken stapsgewijze

procedures te ontwikkelen, die gebaseerd zijn op een nauwkeurige analyse van de concepten en doelstellingen van de technieken. We noemen zo'n procedure voor een bepaalde techniek een *modelleringsprocedure*. We hebben we daarbij de mogelijkheid de ervaring en inzicht van gebruikers en onderzoekers te verwerken in deze procedures. Deze benaderingswijze is voor een deel gebaseerd op het werk rond NIAM (Nijssen e.a. 1989). Andere modelleringsprocedures zijn te vinden in Aerts e.a. (1990) en Brinkkemper (1988-90ab). In deze werken worden deze procedures gecombineerd met een wiskundige aanpak.

Om onze aanpak te illustreren nemen we in dit artikel de gegevensmodellering met behulp van de Entiteit-Relatie techniek als voorbeeld. We hebben de beschrijving als volgt opgebouwd. Allereerst plaatsen we gegevensmodellering in de context van systeemontwerp, om de precieze relaties met de andere gebruikte technieken te bepalen. Procesmodellering, met behulp van bijvoorbeeld Dataflow-diagrammen levert een inzicht op in de activiteiten en informatiestromen en hun onderlinge relatie, die zullen dienen als uitgangspunt voor de gegevensmodellering. Vervolgens wordt er een modelleringsprocedure voor de Entiteit-Relatie modelleringstechniek uitgewerkt door eerst de concepten van deze techniek te definiëren en vervolgens de modelleringstappen te presenteren. Daarna volgt een discussie van het meta-model van de Entiteit-Relatie techniek, te zamen met een aantal formele regels en eigenschappen. We besluiten met enkele consequenties van modelleringsprocedures en formalisatie voor het ontwerpen van informatiesystemen.

2 De context van gegevensmodellering

Gegevensmodellering maakt deel uit van de ontwerpfase van een ontwikkelproject van een informatiesysteem. Zo'n informatiesysteem functioneert als een deel van de informatieverwerking binnen een bedrijf of organisatie, hetgeen bij de projectstart nauwkeurig omschreven is in termen van begrenzing, betrokken personen en delen van de organisatie, vereiste hardwarevoorzieningen, enzovoorts. De methodische aanpak voor systeemontwikkeling bestaat uit een uitgebreid plan van ontwikkelactiviteiten met de bijbehorende producten, waarbinnen we ons voor het doel van dit artikel richten op de directe context van gegevensmodellering, zoals geïllustreerd in figuur 1. Merk hierbij overigens op, dat de aard van de methode en het project kunnen afwijken, waardoor andere contexten mogelijk zijn.



Figuur 1
Context van gegevensmodellering

We zullen nu puntsgewijs de onderdelen uit deze figuur verder uitwerken.

1. Specificatie van informanten

Informanten zijn personen die voldoende kennis hebben over het toepassingsgebied van het te ontwikkelen informatiesysteem. Zij verschaffen informatie over de informatieverwerking binnen het toepassingsgebied. Het verdient aanbeveling om dit in termen van het toepassingsgebied te doen, dus bijvoorbeeld met behulp van brieven, formulieren, rapporten etc., die binnen het toepassingsgebied circuleren. Niet alleen eindgebruikers kunnen informanten zijn, maar ook hun managers en experts in het toepassingsdomein

2. Organisatie activiteiten modelleren

Met behulp van de specificaties van de informanten gaan we nu de activiteiten en de informatiestromen binnen de organisatie nader analyseren. Allereerst onderscheiden we welke activiteiten binnen een organisatie informatie verwerken. Voor elke activiteit zal aangegeven dienen te worden welke personen verantwoordelijk zijn voor de uitvoering daarvan. Om de activiteiten goed uit te kunnen

voeren, hebben deze personen informatie nodig die het resultaat is van andere activiteiten binnen of buiten de organisatie. Zij leveren eventueel op hun beurt weer informatie die nodig is voor andere activiteiten. Dit alles wordt gemodelleerd in het activiteitendiagram.

Het is gewenst om activiteiten binnen een organisatie op vergelijkbare abstractieniveaus te brengen, waarbij de informatiestromen tussen de verschillende activiteiten bestaan uit memo's, brieven, formulieren etc. Indien tussen activiteiten informatiestromen lopen die bestaan uit onderdelen van die brieven, memo's etc., dan dienen deze activiteiten en informatiestromen samengevoegd te worden in één superactiviteit. Indien tussen activiteiten informatiestromen lopen die bestaan uit een combinatie van brieven, memo's etc., dan dienen deze activiteiten en informatiestromen opgesplitst te worden in subactiviteiten. Dit decompositieproces kan gestopt worden indien er een compleet overzicht van de informatieverwerking in het toepassingsgebied verkregen is en er van alle informatiestromen voorbeelden vergaard kunnen worden. Het aantal niveau's van de hiërarchie zal, afhankelijk van de complexiteit van het informatiesysteem, ongeveer 2 tot 4 bedragen.

3. Activiteitendiagram

De resultaten van het modelleren van organisatieactiviteiten worden samengevoegd in een activiteitendiagram. Veelal beslaat dit model meerdere diagrammen, die geordend zijn volgens de hiërarchie van activiteiten. Veelgebruikte notaties hiervoor zijn ISAC A- en I-graphs en Dataflow-diagrammen

4. Gegevensmodelleren

De informatiestromen die in het activiteiten-diagram zijn gevonden worden nu gebruikt, te zamen met additionele specificaties van de informanten, als basisgegevens voor de gegevensmodelleringstechniek. Op elke informatiestroom wordt deze techniek toegepast. In de volgende paragraaf zullen we hier uitgebreid op ingaan.

5. Gegevensmodel

De resulterende gegevensmodellen per informatiestroom worden geïntegreerd tot een of meerdere gegevensmodellen, corresponderend met de databases die in het informatiesysteem voorzien zijn. Notaties voor gegevensmodellen zijn NIAM-diagrammen en Entiteit-Relatie-diagrammen.

6. De detaillering activiteiten model

Met complete gegevensmodellen bij de hand kan het activiteitendiagram nader gedetailleerd en eventueel aangepast worden. De activiteiten worden

geformuleerd in termen van entiteiten en relaties uit het gegevensmodel. Deze modellering wordt voortgezet totdat een voldoende niveau bereikt is om tot (pseudo)codering over te gaan.

7. Gedetailleerd activiteiten diagram

Voor dit diagram kan de notatie van het globale activiteiten diagram gebruikt blijven worden, ofwel wordt overgegaan op een notatie die fijnere procesdetails (iteraties, keuzes, foutafhandeling, database-access) kan weergeven, zoals Action diagrams of Petri-netten.

Het is essentieel de modelleringstechniek correct te positioneren binnen de systeemontwikkelingsmethode en de relaties met de andere activiteiten en producten goed vast te leggen. Beschikbare projectdocumenten kunnen benut te worden als invoer voor de modellering, terwijl de techniek op haar beurt modellen oplevert, die elders in het traject nodig zijn.

3 Gegevensmodellering

De modellering van informatiestromen kan met behulp van verschillende gegevensmodelleringstechnieken uitgevoerd worden, zoals bijvoorbeeld NIAM, het Functioneel datamodel en Entiteit-Relatiediagrammen. We behandelen hier de modelleringprocedure die leidt tot Entiteit-Relatie-diagrammen. We zullen daarvoor eerst de basisconcepten van deze techniek bespreken.

3.1 Concepten van de Entiteit-Relatie modelleringstechniek

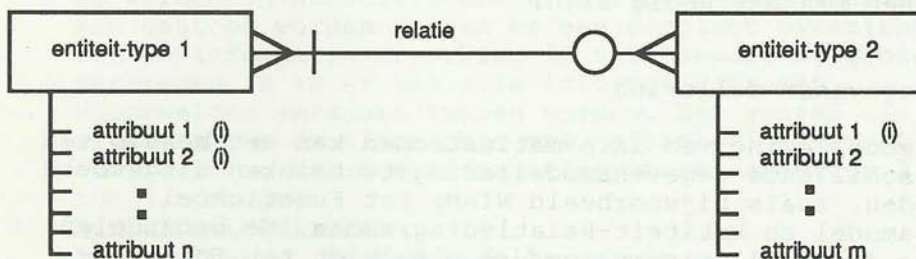
Deze techniek wordt gebruikt om gegevens betreffende objecten, ofwel entiteiten, en hun onderlinge relaties te modelleren. We maken hier gebruik van de eenvoudige binaire versie van deze techniek voor de eenvoud van deze introductie. Extensies aan de modelleringprocedure voor uitgebreidere versies kunnen op basis van het navolgende gerealiseerd worden.

De Entiteit-Relatie modelleringstechniek onderscheidt de volgende concepten:

- **Entiteit-type:** Dit is een verzameling van entiteiten die onder één noemer vallen. Een entiteit is dus een instantie van een entiteit-type.
- **Attributen:** Een attribuut geeft de waarde van eigenschappen van entiteiten aan. Elk attribuut hoort bij precies één entiteit-type en er kunnen meerdere

- attributen voorkomen bij een entiteitstype.
- **Identificatie-structuur:** Dit is een combinatie van één of meerdere attributen, welke een entiteitstype uniek identificeert. De identificatie-structuur wordt ook wel sleutel genoemd.
 - **Relaties:** Dit is een associatie tussen twee (eventueel dezelfde) entiteitstypen.
 - **Cardinaliteiten:** Hiermee geven we het minimum aantal keren (nul of één) en het maximum aantal keren (één of meerdere) dat entiteiten kunnen voorkomen in een bepaalde relatie.

Deze concepten kunnen grafisch weergegeven worden zoals in figuur 2.



Figuur 2

Grafische weergave van gegevensmodelleringsconcepten

De identificatie-structuur wordt aangegeven door achter de attributen die de identificatie-structuur vormen een (i) te plaatsen.

De cardinaliteit van een relatie kan worden aangegeven door middel van drie tekens: 0 voor nul; 1 voor één; < voor meerdere. Het teken dat het dichtst bij het entiteitstype staat geeft de maximum cardinaliteit aan, het andere teken de minimum cardinaliteit.

3.2 Modelleringsprocedure voor Entiteit-Relatie-diagrammen

Het is vrij moeilijk om uitsluitend met kennis van de hierboven behandelde concepten van de Entiteit-Relatie modelleringstechniek een correct Entiteit-Relatie-diagram te construeren. Daarom geven we een modelleringsprocedure bestaande uit zes stappen. Het voorbeeld is simpel gelaten voor de eenvoud van de discussie. Bij de behandeling van deze zes stappen gebruiken we als voorbeeld een kassabon van een

supermarkt. We gaan er vanuit dat er een activiteiten diagram van het beoogde informatiesysteem voor de supermarkt gecreëerd is, waarin een kassabon als informatiestroom voorkomt van de kassier naar de klant.

Stap 1: Verzamelen van voorbeelden.

Voorbeelden die worden gebruikt kunnen bestaan uit formulieren, tabellen, memo's, brieven, rapporten of, zoals in ons voorbeeld, een kassabon. De kassabon die wij gebruiken ziet er uit als in figuur 3 getoond.

AH Daalseweg 173 Nijmegen	
29/05/90	15.56
appelmoes	1.39
pizza	3.98
ijskaart	3.98
v. melk	1.32
	10.67
totaal	10.65
betaald	25.00
terug	14.35
004 1438	

Figuur 3
Voorbeeld van een kassabon

Stap 2: Behandeling van redundantie.

In de voorbeelden vergaard in stap 1 kan dezelfde informatie direct of indirect meerdere malen voorkomen. Dit wordt *redundantie* genoemd. Deze redundantie is niet altijd gewenst vanwege de eventuele negatieve invloed op geheugengebruik en efficiëntie in het informatiesysteem. We zullen daarom redundante gegevens over het algemeen weglaten.

We onderscheiden vier vormen van redundantie:

- Redundantie door herhaling:
Gegevens die meerdere keren voorkomen in het voorbeeld.
- Redundantie door eenvoudige afleiding:
Gegevens die op eenvoudige wijze te berekenen of af te leiden zijn uit andere gegevens, zoals totalen, minima

en maxima, rangnummers.

- Redundantie door meer gecompliceerde afleiding:
Gegevens die afgeleid kunnen worden uit een combinatie van andere gegevens en waarvoor complexe berekeningen nodig zijn. Een voorbeeld hierbij is in Nederland de straatnaam, die in principe afleidbaar is uit de postcode en het huisnummer.
- Redundantie door gedeeltelijke afleiding:
Gegevens die voor een deel afgeleid kunnen worden uit andere gegevens, maar waarvoor uitzonderingen bestaan. Een voorbeeld hierbij is het jaar van aanvang van de studie van een student, dat afleidbaar is uit het studentnummer. Voor studenten die een tweede studie volgen is dit echter niet het geval.

In het voorbeeld in figuur 3 zijn de items "totaal" en "terug" redundant door eenvoudige afleiding. Ook het niet afgeronde totaal is afleidbaar.

Stap 3: Typering van objecten.

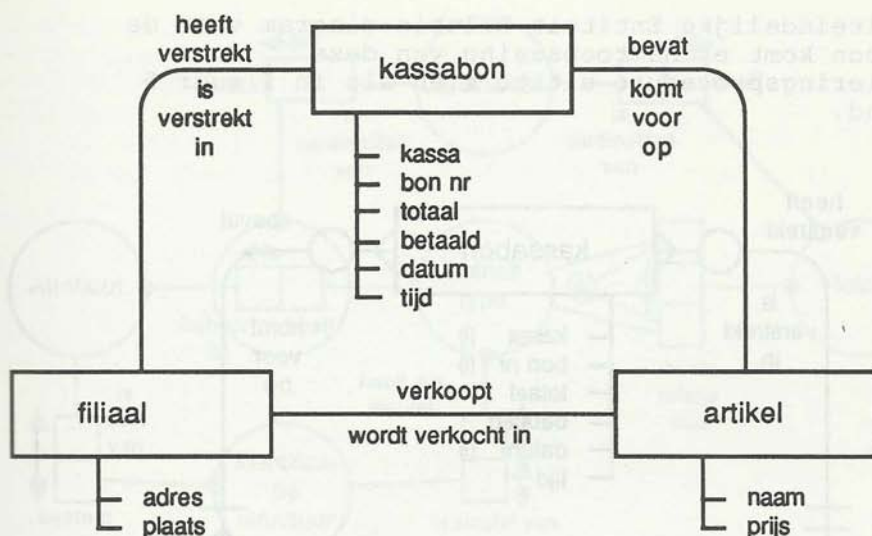
Na het verwijderen van redundantie vindt de typering van objecten in attributen, entiteit-typen en relaties plaats.

- Bepaling van de attributen
Attributen zijn de meest elementaire eenheden van gegevens. Ze hebben vaak een concrete vorm in de voorbeelden of dienen ingevuld te worden in het geval dat het voorbeeld een formulier is.
Op de kassabon zijn het onder andere 'straatnaam', 'prijs' en 'totaal'
- Koppeling van attributen aan entiteit-typen
Elk van de hierboven gevonden attributen dienen te worden gegroepeerd rond entiteit-typen.
In ons voorbeeld zijn dat 'kassabon', 'artikel' en 'filiaal'.
Het entiteit-type 'artikel' krijgt zo de attributen 'naam' en 'prijs'
- Koppeling van de entiteit-typen
Entiteit-typen moeten via een aantal relaties aan minstens één andere entiteit-type gekoppeld worden. Deze relaties krijgen daarbij twee namen voor beide richtingen van de relatie.
De relatie tussen 'artikel' en 'filiaal' krijgt de namen 'verkoopt' en 'wordt verkocht in'.

Voor de kassabon verkrijgen we een voorlopig Entiteit-Relatie diagram als in figuur 4.

Stap 4: Bepalen van cardinaliteiten.

De cardinaliteiten voor de relaties tussen entiteit-typen kunnen meestal uit de voorbeelden worden gehaald. Is dit niet het geval, dan kan de informant hierover



Figuur 4
Entiteit-Relatie-diagram na stap 3

meer informatie verstrekken. De informant kan in ons voorbeeld de regel 'Een kassabon bevat minimaal één artikel' beamen en de regel 'Een kassabon bevat maximaal één artikel' ontkennen.

Stap 5: Bepalen van naamgevingsconventies.

Een universele, algemeen geaccepteerde terminologie in een toepassingsgebied is zeldzaam. Veelal gebruiken verschillende personen verschillende termen voor hetzelfde begrip. Om duidelijkheid te bevorderen en om fouten te vermijden, dienen er afspraken gemaakt te worden voor alle gegevensobjecten van het informatiesysteem.

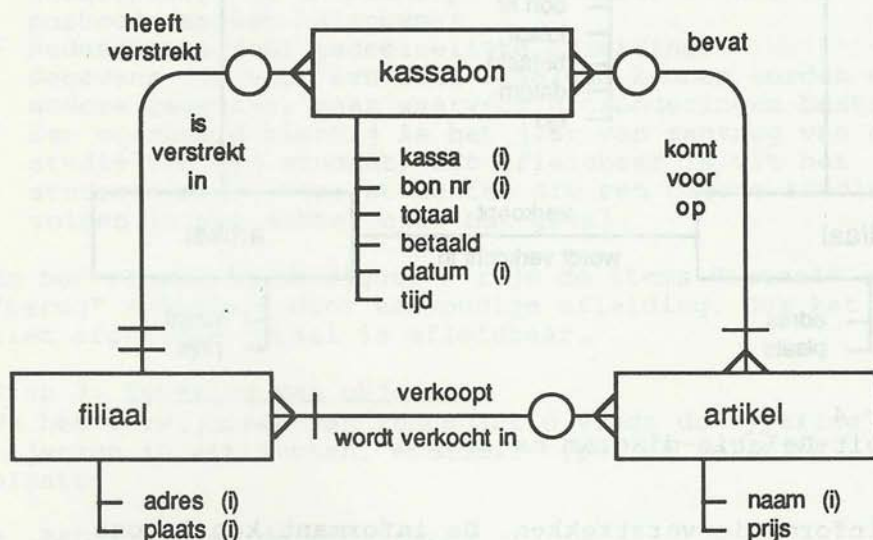
In ons voorbeeld zou het kunnen zijn dat in plaats van de term 'artikel' ook de term 'product' gebruikt wordt. Er dient dan afgesproken te worden dat in het vervolg alleen 'artikel' gebruikt zal worden.

Stap 6: Bepalen van de sleutels voor entiteit-typen.

In deze laatste stap wordt een identificatie-structuur bepaald voor elk entiteit-type. Dit wordt gedaan door te bepalen welke minimale combinatie van attributen uniek is bij alle entiteiten van een entiteit-type. Het is mogelijk dat er alternatieve identificatie-structuren bestaan voor een entiteit-type.

In ons voorbeeld is de combinatie van het nummer van de kassa, het nummer van de bon en de datum uniek voor elke kassabon.

Het uiteindelijke Entiteit-Relatie-diagram voor de kassabon komt er na toepassing van deze modelleringsprocedure uit te zien als in figuur 5 getoond.

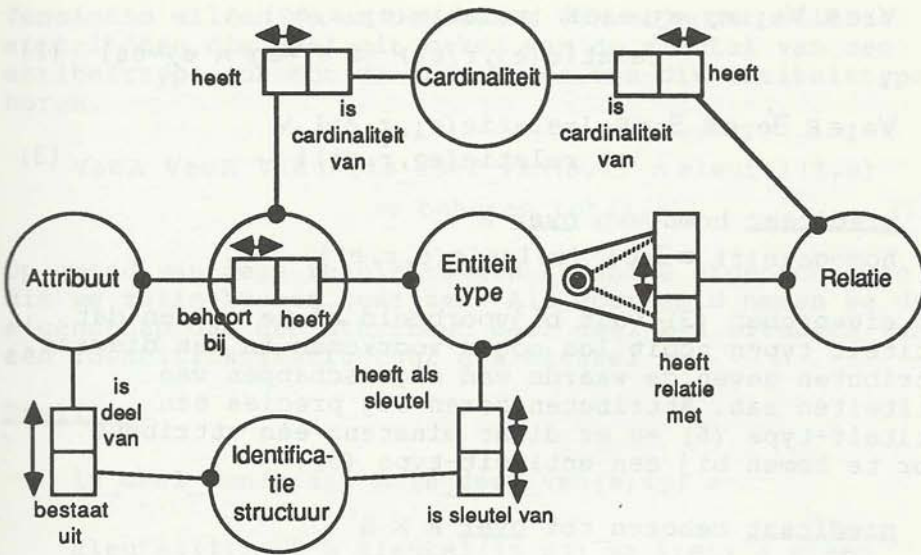


Figuur 5
Het uiteindelijke Entiteit-Relatie-diagram

4 Meta-modellering en formalisatie

We kunnen ons nu afvragen hoe we modelleringsprocedures kunnen construeren voor een willekeurige techniek. Tevens willen we in staat zijn theorie over modelleringstechnieken op te bouwen, die we kunnen integreren in het modelleringsproces. Dit blijkt door middel van meta-modellering en wiskundige formalisatie heel goed mogelijk te zijn. We zullen dit in deze paragraaf toelichten aan de hand van de Entiteit-Relatie modelleringstechniek.

Allereerst tonen we van deze techniek het meta-model in figuur 6 in de NIAM notatie. De concepten, zoals ze in paragraaf 3.1 behandeld werden, staan in deze techniek genoteerd in cirkels en de onderlinge relaties als aaneengesloten blokjes. Merk nu op dat deze concepten in de procedure van paragraaf 3.2 allemaal voorkomen: bijv. Attribuut in stap 3 en Cardinaliteit in stap 4. Zo ook worden de relaties tussen de concepten in een van de stappen behandeld: bijv. Entiteit-type heeft Identificatie-structuur in stap 6. Tenslotte dienen ook alle constraints in het meta-model in de procedure ergens tot een controle stap te leiden.



Figuur 6
Het meta-model van de Entiteit-Relatie
modelleringstechniek

Van dit meta-model naar een wiskundige formalisatie middels predicaten logica is niet zo'n moeilijke stap. De constructies van deze logica en de mogelijkheden tot het bewijzen van stellingen kunnen we benutten in de toepassing van de techniek en bij het bouwen van hulpmiddelen voor modelleringstechnieken. We zullen dit illustreren door middel van enige eigenschappen van de Entiteit-Relatie techniek. Hiertoe introduceren we de volgende verzamelingen voor de basisconcepten van de Entiteit-Relatie modelleringstechniek:

- E: de verzameling van entiteit-typen
- R: de verzameling van relaties
- A: de verzameling van attributen
- I: de verzameling van identificatie-structuren
- C: de verzameling van cardinaliteiten

Entiteit-typen zijn gerelateerd aan elkaar door middel van verschillende binaire relaties. Homogene relaties, dat wil zeggen relaties van entiteit-typen naar zichzelf, zijn ook toegestaan. Deze aspecten uit het meta-model kunnen getransformeerd worden in de predicaten logica.

predicaat relatie over $E \times R \times E$

$$\forall r \in R \exists e_1, e_2 \in E [\text{relatie}(e_1, r, e_2)] \quad (1)$$

$$\forall r \in R \quad \forall e_1, e_2, e_3, e_4 \in E \quad [\text{relatie}(e_1, r, e_2) \wedge \text{relatie}(e_3, r, e_4) \Rightarrow e_1 = e_3 \wedge e_2 = e_4] \quad (2)$$

$$\forall e_1 \in E \quad \exists e_2 \in E \quad \exists r \in R \quad [\text{relatie}(e_1, r, e_2) \vee \text{relatie}(e_2, r, e_1)] \quad (3)$$

predicaat homogeen over R

$$\text{homogeen}(r) \equiv \exists e \in E \quad [\text{relatie}(e, r, e)]$$

Uit eigenschap (3) valt bijvoorbeeld af te leiden dat entiteit typen nooit los mogen voorkomen in een diagram. Attributen geven de waarde van eigenschappen van entiteiten aan. Attributen horen bij precies één entiteit-type (6) en er dient minstens één attribuut voor te komen bij een entiteit-type (5).

predicaat behoren_tot over A × E

$$\forall a \in A \quad \exists e \in E \quad [\text{behoeren_tot}(a, e)] \quad (4)$$

$$\forall e \in E \quad \exists a \in A \quad [\text{behoeren_tot}(a, e)] \quad (5)$$

$$\forall a \in A \quad \forall e_1, e_2 \in E \quad [\text{behoeren_tot}(a, e_1) \wedge \text{behoeren_tot}(a, e_2) \Rightarrow e_1 = e_2] \quad (6)$$

De identificatie-structuur bestaat uit attributen, die een entiteit-type uniek identificeren, ofwel de sleutel vormen. Een entiteit type kan hierdoor meerdere identificatie-structuren hebben, waarvan er een als sleutel gekozen is. Voor ieder entiteit-type geldt dat deze precies één sleutel heeft (8).

predicaat sleutel over I × E

predicaat is_deel_van over A × I

$$\forall e \in E \quad \exists i \in I \quad [\text{sleutel}(i, e)] \quad (7)$$

$$\forall i \in I \quad \forall e_1, e_2 \in E \quad [\text{sleutel}(i, e_1) \wedge \text{sleutel}(i, e_2) \Rightarrow e_1 = e_2] \quad (8)$$

$$\forall e \in E \quad \forall i_1, i_2 \in I \quad [\text{sleutel}(i_1, e) \wedge \text{sleutel}(i_2, e) \Rightarrow i_1 = i_2] \quad (9)$$

$$\forall i \in I \quad \exists a \in A \quad [\text{is_deel_van}(a, i)] \quad (10)$$

Tenslotte willen we er ook zeker van zijn dat alle attributen die deel uit maken van de sleutel van een entiteitstype ook tot de attributen van dit entiteitstype horen.

$$\forall a \in A \quad \forall e \in E \quad \forall i \in I \quad [is_deel_van(a, i) \wedge sleutel(i, e) \Rightarrow behoren_tot(a, e)] \quad (11)$$

Op grond van deze regels volgen er enige eigenschappen die we zelfs kunnen bewijzen. Als voorbeeld nemen we de eigenschap dat een attribuut maar deel kan uitmaken van één identificatiestructuur die sleutel is.

Stelling:

$$is_deel_van(a, i_1) \wedge is_deel_van(a, i_2) \wedge sleutel(i_1, e_1) \wedge sleutel(i_2, e_2) \Rightarrow i_1 = i_2 \wedge e_1 = e_2$$

Bewijs:

Uit het gegeven volgt met behulp van regel (11):

$$behoren_tot(a, e_1) \wedge behoren_tot(a, e_2)$$

Volgens (6) geldt dan $e_1 = e_2$

We hebben dan $sleutel(i_1, e_1) \wedge sleutel(i_2, e_1)$,

waaruit volgens (9) $i_1 = i_2$ volgt.

Op een analoge wijze zijn ook de andere relaties en regels uit het meta-model te formaliseren met behulp van de predicaten logica. Wij zullen dit hier niet uitwerken; dit wordt als oefening aan de lezer overgelaten.

Merk tenslotte nog op, dat hulpmiddelen voor een modelleringstechniek als bijvoorbeeld Entiteit-Relatie diagrammen, de regels implementeren als consistentiecontroles op de diagrammen. De afleidbare eigenschappen zijn dan automatisch gewaarborgd.

5 Afsluiting

Het modelleren van informatiesystemen kan aanzienlijk in kwaliteit vooruitgaan door een meer formele aanpak en het expliciteren van de stappen van de techniek in de modelleringsprocedures.

Dit komt onder meer tot uiting in:

Verbeterde modellering

De intuïtieve aanpak van modellering kan door de beschikbaarheid van een modelleringstheorie verdwijnen. Daarvoor in de plaats komt er een meer gestructureerde formele aanpak.

Betere ontwerper-informanten dialoog

De communicatie-kloof tussen de informanten en de ontwerpers kan overbrugd worden door voorschriften te geven voor de informatie die nodig is van de informanten. Daarbij kan de informant gebruik maken van de beschikbare documenten uit het toepassingsdomein en van zijn eigen terminologie. De informant behoeft dus niet te communiceren in termen die alleen bekend zijn aan de ontwerper.

Conceptuele duidelijkheid

De formulering van de concepten van een modelleringstechniek in een wiskundig model is er op gericht een precies begrip van de concepten te verkrijgen. Er kan hierdoor extra kennis over de modelleringstechniek verkregen worden.

Automatiseerbaar

Sommige stappen kunnen geheel of gedeeltelijk geautomatiseerd worden dankzij de formele beschrijving van modellering.

De in dit artikel besproken aanpak om tot een modelleringsprocedure van de Entiteit-Relatie techniek te komen, kan zonder meer ook toegepast worden op andere modelleringstechnieken, zoals Dataflow diagrammen en Structure Charts. Ons is gebleken dat er op de keper beschouwd nog vrij weinig theorie voorhanden is over deze technieken, die voor een goede toepassing echter noodzakelijk is en dus ontwikkeld moet worden. Voor een deel wordt dit op het ogenblik gedaan door de bouwers van hulpmiddelen van deze tools. Maar gezien de uiteenlopende kwaliteit van de implementaties ligt er op het gebied van het modelleren van informatiesystemen nog veel terrein braak.

Gebruikte literatuur

- Aerts, A.T.M. and K.M. van Hee (1989) Modelleren met een functioneel datamodel. Informatie, vol. 31, pp. 941-956.
- Brinkkemper, S., N.A. Brand and J.Moormann (1988) Deterministic Modelling Procedures for Automated Analysis and Design Tools. In: T.W. Olle, A.A. Verrijn Stuart and L.Bhabuta (Eds.), Computerized Assistance during the Information Systems Life Cycle, Proceedings of the CRIS 88 conference, North-Holland, Amsterdam,

pp. 117-160, 1988. Extended Version in Technical report nr. 88-10, Department of Information Systems, University of Nijmegen, May 1988

Brinkkemper, S. (1990a). Formalisation of Information Systems Modelling. Dissertation, University of Nijmegen, the Netherlands, Thesis Publishers, Amsterdam

Brinkkemper, S., L.J.M. Gerritsen and J.E.A. van Hintum (1990b) Het toepassen van modelleringstechnieken als Dataflow-diagrammen en Entiteit-Relatie-diagrammen. Technical report, Department of Information Systems, University of Nijmegen

Nijssen, G.M. and T.A.Halpin (1989) Conceptual Schema and Relational Database Design: a Fact-Based Approach. Prentice Hall