

Inleiding Modelleren in de Bestuurlijke Informatiekunde

H.W.M.Gazendam en S.Sibum

Faculteit Bedrijfskunde
Rijksuniversiteit Groningen
Groningen, 10 maart 1997

Inhoudsopgave

1. Inleiding: Modelleren in de informatiekunde	4
1.1 Informatiesystemen en modelleren.....	4
1.2 Abstractie en reticulatie.....	4
1.3 Soorten handelingen.....	6
1.4 Soorten modellen.....	6
1.5 Een methode voor informatiekundig modelleren.....	6
1.6 Diagrammen.....	7
1.7 Het gebruik van System Architect.....	8
2. Informatiesysteem en informatiekunde.....	11
2.1 Informatiesysteem.....	11
2.2 Informatievoorziening en informatiehuishouding, automatisering en informatisering.....	12
2.3 Het vakgebied bestuurlijke informatiekunde.....	13
3. Organisatie en informatiesysteem.....	15
3.1 Typen informatiesystemen.....	15
3.2 Het gebruik van informatietechnologie.....	16
3.3 Verantwoordelijkheden op het gebied van informatiesystemen.....	17
3.4 De functionaliteit van het informatiesysteem.....	18
3.5 Het gebruikersinterface van het informatiesysteem.....	19
3.6 De opbouw van het informatiesysteem.....	20
4. Organisatie en interactie.....	22
4.1 Actor.....	22
4.2 Programma, impuls, actie, activiteit, proces.....	22
4.3 Organisatie.....	23
4.4 Zaak, prestatie.....	23
4.5 Subjectsysteem en objectsysteem.....	24
4.6 Interactie, overdrachtsactie, communicatieve actie.....	24
4.7 Algemene structuur van de interactie, actagene en factagene conversatie.....	25
4.8 Transactie.....	26
4.9 Transformatie.....	27
5. Objecttype en associatietype.....	28
5.1 Object, begrip, symbool.....	28
5.2 Objecttype en attribuuttype.....	29
5.3 Associatie, associatietype, generalisatie.....	29
5.4 Structuren.....	32
5.5 Categorieën.....	33
6. Verbale beschrijving van organisatie en informatiesysteem.....	36
6.1 Overzicht van te beschrijven aspecten van organisatie en informatiesysteem.....	36
6.2 Typen informatiesystemen.....	36
6.3 Houding van de organisatie tegenover informatietechnologie.....	37

6.4 De verantwoordelijkheidsverdeling op het gebied van informatiesystemen.....	38
6.5 Typologie Starreveld	38
6.6 Organigram.....	38
6.7 Bedrijfsprocessen	39
6.8 Gegevensstromen en bestanden.....	40
6.9 De functionaliteit van het informatiesysteem.....	41
6.10 Het gebruikersinterface van het informatiesysteem	42
6.11 De opbouw van het informatiesysteem	43
7. Actor-interactiemodellen (B1-modellen)	45
7.1 Actor-interactiemodellen (B1- modellen)	45
7.2 Organisaties en transacties	45
7.3 Actoren, transacties en transformaties.....	47
8. Bedrijfsprocesmodellen (B2-modellen)	50
8.1 Bedrijfsprocesmodellen (B2-modellen)	50
8.2 Verbanden tussen interacties	50
8.3 Tijdslijnen van actoren en acties	53
8.4 De interne structuur van een interactie.....	55
8.5 Beoordeling van bedrijfsprocessen	56
9. Documentmodellen (D-modellen).....	58
9.1 Documentmodellen (D-modellen).....	58
9.2 Uitwerking interacties in termen van activiteiten en berichten.....	58
9.3 Actoren en berichten	59
9.4 Beoordeling van het informatiegebruik door actoren.....	61
10. Objectmodellen (I-modellen)	63
10.1 Objectmodellen (I-modellen)	63
10.2 Objecttypen	64
10.3 Associatietypen	65
10.4 Objectmodel	67
11. Case Fietsenhandel JEEKAA.....	71
11.1 Omschrijving	71
11.2 Bestelformulier	72
11.3 Factuur.....	73
12. Literatuur.....	74
13. Bijlage 1. Typologie Starreveld	76

1. Inleiding: Modelleren in de informatiekunde

1.1 Informatiesystemen en modelleren

Informatiesystemen bevatten modellen van de werkelijkheid. Die modellen gaan meestal over organisaties. Daarbij is van belang dat organisaties met andere organisaties interacteren, en dat organisaties intern functioneren. In de manier waarop dat gebeurt zijn regelmatigheden in de vorm van structuren en processen te onderkennen die de basis vormen van de modellen die in informatiesystemen worden opgeslagen. Modelleren van organisaties is daarom een kern van het vak bestuurlijke informatiekunde. Van de manier waarop we de werkelijkheid modelleren hangt immers af hoe het informatiesysteem dat op grond van dat model is gebouwd zich zal gedragen. Modelleren is een onderwerp waarover een bedrijfskundige met een technicus moet kunnen praten, omdat bij het modelleren vorm wordt gegeven aan de manier waarop organisaties werken, netwerken van organisaties zich gedragen, wet- en regelgeving wordt uitgevoerd, en besturing greep krijgt op te besturen processen en actoren.

1.2 Abstractie en reticulatie

Bij het modelleren gekozen beschouwingsniveaus kunnen op twee manieren van elkaar verschillen: in de mate van *reticulatie*, d.w.z. de fijnmazigheid waarmee wordt gekeken, en in de mate van *abstractie*, d.w.z. het kijken naar algemene eigenschappen en het afzien van individuele gevallen.

Reticulatie is het proces waarbij men meer fijnmazig gaat waarnemen (De Leeuw, 1988: 105). Dereticulatie is het proces waarbij men minder fijnmazig gaat waarnemen. Als reticulatie delen van een geheel zichtbaar maakt die eerst niet zichtbaar waren spreekt men ook wel van decompositie.

Het *reticulatieniveau* is de mate van fijnmazigheid waarmee een systeem wordt waargenomen en beschreven. Het verhogen van het reticulatieniveau zorgt dat andere verschijnselen kunnen worden waargenomen. Decompositie is een vorm van reticulatie. Bij een *decompositiestap* worden de delen van een geheel beschreven, mogelijk inclusief de relatie tussen die delen. Op die delen kan vervolgens weer een decompositiestap worden toegepast, enzovoorts. Het aantal decompositiestappen dat kan worden toegepast is onder meer afhankelijk van het bestaan van een laagste, ondeelbaar niveau (bijvoorbeeld de mens in een organisatie). Het reticulatieniveau wordt in modellen vaak geteld als het aantal decompositiestappen dat bij een beschrijving is genomen vanaf het ongedeelde systeem (dat is dus 0, 1, 2, etc.).

Voorbeeld: men begint bij het waarnemen van het systeem **Rijksuniversiteit Groningen** (reticulatieniveau 0). Daarna past men een decompositiestap toe en ziet de subsystemen **Bestuur**, **Bureau**, en **Faculteit A**, **Faculteit B**, **Faculteit C**, . . . (reticulatieniveau 1). Vervolgens verhoogt men het reticulatieniveau, maar zonder decompositie toe te passen, en ziet dat er geldstromen lopen vanuit het **Bureau** naar de **Faculteiten** (reticulatieniveau 1a). Verhoogt men het reticulatieniveau nog eens, dan ziet men bijvoorbeeld ook de besturingsrelaties tussen **Bestuur**, **Bureau** en **Faculteiten** (reticulatieniveau 1b).

Abstractie is het proces waarbij men de algemeen geldende eigenschappen van entiteiten in beschouwing neemt en afziet van identificerende en toevallige eigenschappen van individuele entiteiten. Een verzameling bij elkaar horende eigenschappen noemen we een *type*.

Bijvoorbeeld: men abstraheert van de identificerende en toevallige eigenschappen van de personen **A.Alberts** en **E.Egberts** en spreekt over de 'student' die verplicht is om jaarlijks 21 studiepunten te halen. STUDENT is dan het type, iets waarover men spreekt op een hoger abstractieniveau in plaats van over de specifieke **A.Alberts** en **E.Egberts** die beiden ook als student zijn ingeschreven en derhalve ook 21 studiepunten moeten halen.

Concretisatie is het proces waarbij individuele entiteiten die behoren bij een type in ogenschouw worden genomen. Een type kan zelf ook weer worden beschouwd als een entiteit met eigenschappen zodat een type type mogelijk wordt; een type type noemen we een *categorie*.

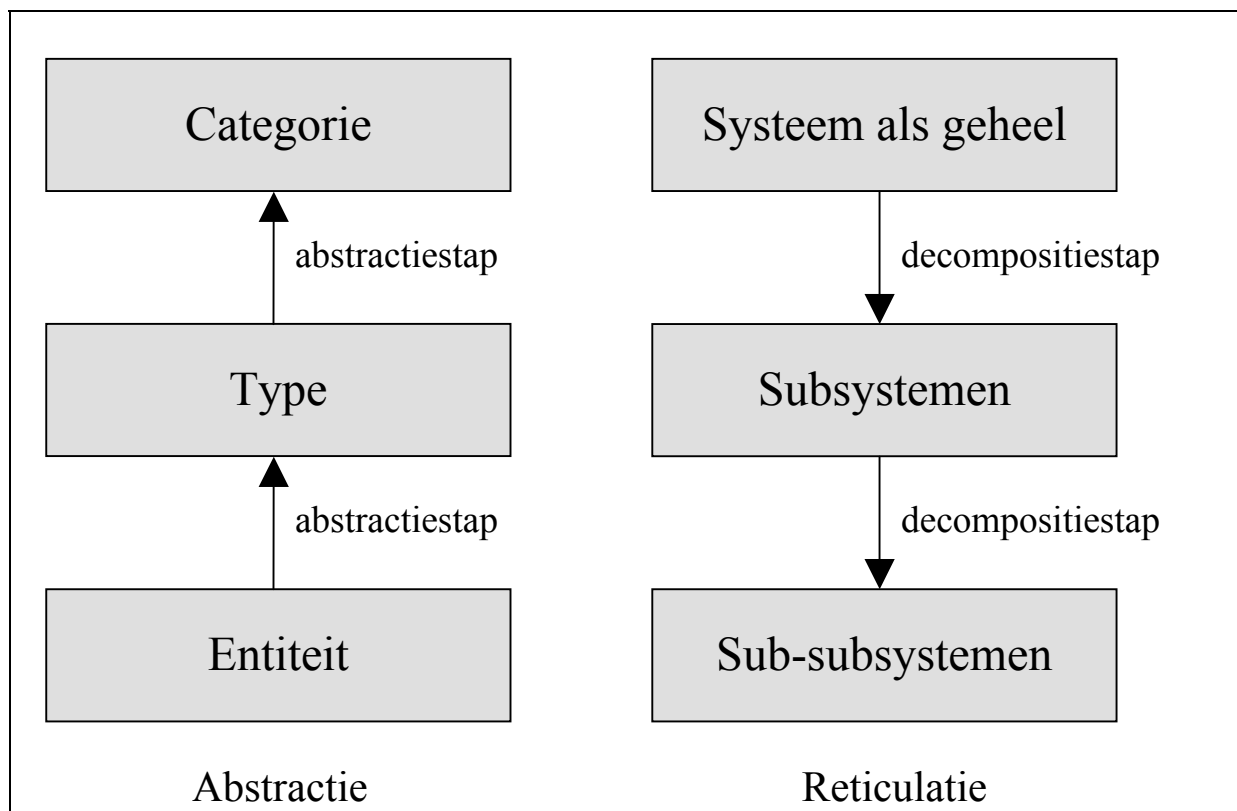
Voorbeeld:

Als men uitgaat van het type STUDENT en men geeft daar het voorbeeld bij van de student **A.Alberts** is dat concretisatie. Als men uitgaat van het type STUDENT en men zegt dat STUDENT behoort tot de actoren, dan heeft men het over de categorie *actor*.

We onderscheiden drie abstractieniveaus:

- het niveau der entiteiten;
- het niveau der typen;
- het niveau der categorieën.

De modellen die wij zullen maken liggen allemaal op het type-abstractieniveau.



1.3 Soorten handelingen

Handelingen kunnen worden onderscheiden in:

- essentiële handelingen: alle handelingen waar geen taal aan te pas komt;
- taalhandelingen: handelingen waar taal aan te pas komt.

Binnen de *essentiële handelingen* kunnen we onder meer onderscheiden:

- overdrachtshandelingen waarbij een product of dienst door een actor aan een andere actor wordt overgedragen (deze overdracht kan zowel fysiek zijn en/of rechten en verplichtingen betreffen);
- omzettingshandelingen waarbij een object wordt gecreëerd of van toestand wordt veranderd.

Als een spreker een zin uit, voert hij tegelijkertijd drie soorten *taalhandelingen* uit (Searle, 1969/1977:33):

- hij produceert woorden die samen een zin vormen (uitingshandeling);
- hij verwijst ergens naar en kent daar een eigenschap aan toe (propositionele handeling);
- hij beweert, vraagt, belooft, beveelt, etc. (illocutionaire handeling).

Voorbeeld:

"Geef mij dat boek."

Uitingshandeling:

ik zeg: "geef mij dat boek".

Propositionele handeling:

ik verwijst naar dat rechthoekige voorwerp op tafel en ken daar de eigenschap 'het zijn van een boek' aan toe.

Illocutionaire handeling:

ik maak duidelijk dat ik wil dat jij mij dat boek geeft (van de toon en omstandigheden is afhankelijk of dat een bevel of verzoek is).

1.4 Soorten modellen

Gebaseerd op de nu onderscheiden handelingen kunnen de volgende vier soorten modellen worden onderscheiden (zie ook Dietz, 1996: 4):

- *actor-interactiemodellen* beschrijven essentiële handelingen (B1-modellen van bedrijf);
- *bedrijfsprocesmodellen* beschrijven essentiële handelingen en illocutionaire handelingen, in het bijzonder besturingsgerichte illocutionaire handelingen (B2-modellen van bedrijf en besturing);
- *objectmodellen* beschrijven propositionele handelingen (I-modellen van informatie);
- *documentmodellen* beschrijven uitingshandelingen (D-modellen van documenten).

1.5 Een methode voor informatiekundig modelleren

In deze syllabus wordt het eerste deel van een methode voor informatiekundig modelleren behandeld. Dit eerste deel betreft het beschrijvend modelleren van organisaties en informatiesystemen. Het tweede deel van de methode betreft het modelleren dat is gericht op het (her)ontwerpen van organisaties en informatiesystemen. De gepresenteerde methode is het resultaat van een proces van ontwikkeling tussen 1992 en 1997 (Gazendam en Sibum, 1995).

De stappen in het modelleren die in dit eerste deel aan de orde komen zijn gebaseerd op een synthese van:

- de DEMO methode (Dietz, 1992; 1996);
- de systeemtheoretische/ bedrijfskundige besturingstheorie (De Leeuw, 1988);
- de Role Activity Diagrams (Ould, 1995);
- het semiotisch/ ontologisch modelleren (Gazendam, 1994);
- de objectgeoriënteerde methode OMT (Object Modeling Technique) (Rumbaugh, e.a., 1991) en
- de bedrijfskundig objectgeoriënteerde methode The Object-Oriented Enterprise Model (Gale and Eldred, 1996).

De gebruikte notatie is die van OMT. Deze notatie wordt door het pakket System Architect ondersteund. De praktische gang van zaken bij het tekenen van modelleren zal aan de hand van dit pakket worden behandeld.

Voordat met het eigenlijke modelleren van organisaties kan worden begonnen moet eerst een informatiekundige kijk op de werkelijkheid worden ontwikkeld. Dit gebeurt door het ontwikkelen van een begrippennetwerk dat in vier hoofdstukken zal worden uiteengezet. Achtereenvolgens komen aan de orde de begrippen rond *informatiesysteem en informatiekunde* (Hoofdstuk 2), *organisatie en informatiesysteem* (Hoofdstuk 3), *organisatie en interactie* (Hoofdstuk 4) en *objecttype en associatietype* (Hoofdstuk 5).

Daarna wordt een systematische verbale beschrijving van organisaties en de erin aanwezige informatiesystemen behandeld die de basis vormt van de te maken modellen (Hoofdstuk 6).

Er worden daarna vier typen modellen besproken:

- actor-interactiemodellen (modellen van transacties en transformaties) (Hoofdstuk 7);
- bedrijfsprocesmodellen (modellen van de besturing van transacties en transformaties) (Hoofdstuk 8);
- documentmodellen (modellen van de overdracht en bewaring van berichten, producten, en dergelijke) (Hoofdstuk 9);
- informatiemodellen (modellen van de mogelijke uitspraken die door een informatiesysteem kunnen worden vastgelegd) (Hoofdstuk 10).

De praktische gang van zaken bij het modelleren van organisaties en informatiesystemen wordt uitgelegd aan de hand van de case Fietsenzaak JEEKAA (Hoofdstuk 11).

1.6 Diagrammen

Hieronder volgt een overzicht van de diagrammen die aan de orde zullen komen (tussen haakjes de te gebruiken System Architect module):

Verbale beschrijving:

STRUC organigram (Auto-decomposition)

Actor-interactiemodellen (BI-modellen):

- organisatie en transactie (alleen de overdracht van producten en diensten is zichtbaar; dit is het niveau waarop de transactie in de economie wordt beschreven);

- actor en interactie (de overdracht van rechten, producten, diensten en geld wordt afzonderlijk zichtbaar; dit is het niveau waarop de transactie in de DEMO-theorie wordt beschreven).

Binnen het reticulatieniveau 'actor en interactie' onderscheiden we twee decompositieniveaus waarbij de volgende diagrammen horen:

- ORGAN organisaties en transacties (actor en interactie voor het centrale systeem als geheel) (OMT Data Flow);
- ACTOR actoren, transacties en transformaties (actor en interactie binnen het centrale systeem; hier worden functionele subsystemen zichtbaar gemaakt) (OMT Data Flow).

Bedrijfsprocesmodellen (B2-modellen):

- INTER verbanden tussen interacties (de besturingsacties die interacties met elkaar verbinden worden zichtbaar gemaakt) (OMT State);
- BPROC tijdslijnen van actoren en acties (bedrijfsprocesdiagram) (alle reguliere besturingsacties rond een overdrachtsactie worden zichtbaar) (OMT Event Trace);
- STATE interne structuur van een interactie (naast de reguliere besturingsactie worden nu ook de uitzonderings-besturingsacties zichtbaar) (OMT State).

Documentmodellen (D-modellen):

- DATAF actoren en berichten (OMT Data Flow).

Objectmodellen (I-modellen):

- OBJEC objectmodel (OMT Object).

Figuren krijgen een naam opgebouwd uit twee letters die de gemodelleerde organisatie aangeven, vijf letters die het soort diagram aanduiden en één teken (letter of cijfer) dat de versie of het volgnummer aanduidt.

1.7 Het gebruik van System Architect

Het gebruik van System Architect is grotendeels duidelijk door de manier waarop het gebruikersinterface is vormgegeven.. Er is een redelijke **Help** functie. Enkele aandachtspunten worden hieronder gegeven.

We kunnen diagrammen tekenen met System Architect, en vervolgens het diagram met kopiëren en plakken (**Edit| Select All; Edit| Copy** beide in System Architect en vervolgens: **Edit| Paste** in MS Word) vanuit System Architect naar een tekstverwerker (zoals MS Word) overhalen. Het diagram kan in MS Word eventueel verder worden geëdit.

Door een rechtermuisklik op een symbool krijgen we een pop-up menu te zien dat gaat over de eigenschappen van dat symbool. Erg handig is bijvoorbeeld de eigenschap **Name, Number, Properties** waarmee de naam van het symbool kan worden veranderd. De eigenschap **Addresses** kan worden gebruikt om aan te geven of een symbool betrekking heeft op een Business Objective, Business Process, Critical Success Factor, Geographic Location, etc. Dit is vooral van belang bij hiërarchisch opgebouwde decompositiediagrammen (te tekenen met module Auto Decomposition) die voor allerlei doeleinden (bijvoorbeeld organigram, doelstellingsboom) kunnen worden gebruikt. De eigenschap **Comment** maakt het mogelijk uitleg aan een symbool toe te voegen.

Het toekennen van tweemaal dezelfde naam mag niet in System Architect. Wil men een symbool met een bepaalde naam toch opnemen in een diagram, dan kan dit alsnog gebeuren door het kopiëren en plakken van het originele symbool (meestal vanuit een ander diagram). Het gekopieerde symbool kan eventueel van aard worden veranderd **door <rechtermuisklik>| Transform Symbol**.

Pijlen zitten soms niet goed vast aan een symbool. Ze worden dan grijs. Pijlen kunnen goed worden vastgemaakt door het uiteinde ervan *over* het symbool te trekken. Er verschijnt dan een kruisje.

Soms geeft het automatisch route kiezen van pijlen een verward beeld. Het automatisch route kiezen van pijlen kan worden uitgeschakeld door de optie **Set| Preferences| Immediate Auto-Routing** uit te zetten. Als dit automatisch route kiezen is uitgeschakeld kan door middel van **Edit | Route Now** (of de flits-knop rechts bovenaan) het automatisch route kiezen eenmalig worden uitgevoerd. Het traject van de verbindingspijlen kan na uitschakeling van automatisch route kiezen worden veranderd door de pijl te selecteren en vervolgens de knikpunten te verslepen. Men kan door een **<rechtermuisklik>** op de pijl en vervolgens de optie **Insert Line Segment** een knikpunt toevoegen. Met een **<rechtermuisklik>** op de pijl en **Make Orthogonal** kan men de pijl netjes orthogonaal maken.

Bij het gebruik van het **Rectangle** symbool, bijvoorbeeld om een systeemgrens aan te geven, gaat System Architect zich vaak narrig gedragen. Verbindingslijnen worden bij het automatisch route kiezen dan soms heel vreemd getekend. Het is soms beter deze optie niet te gebruiken en de rechthoek/systeemgrens in MS Word bij te tekenen (bijvoorbeeld als men een grijze systeemgrens wil tekenen zoals in deze syllabus is gebeurd). Een andere mogelijkheid is het automatisch route kiezen van verbindingspijlen uit te schakelen (zie hierboven).

Een overzicht van de al gemaakte diagrammen wordt verkregen door **Diagram| Open| Search**. Het overzicht kan als een aparte window (genaamd Diagrams) open blijven, maar moet na het wissen van een diagram door middel van het **Search** commando weer worden ververs.

Het gebruik van de onderwijsversie van System Architect houdt beperkingen in. Zo kunnen er maximaal 2 projecten, 10 diagrammen en 300 symbolen worden gemaakt. Het is aan te raden zodra deze beperkingen zich laten gelden de volgende stappen te zetten:

- een back-up van de twee directories/folders Project1 en Project2 maken;
- alle gemaakte figuren naar MS-Word kopiëren om ze daar verder te kunnen gebruiken;
- de figuren die niet direct meer nodig zijn wissen.

De volgende opties kan men beter niet gebruiken:

- **Run Report;**
- **OMT Data Flow| OMT Rules Check;**
- **OMT Data Flow| Expression Check;**
- **OMT State| OMT Rules Check;**
- **Definition| Add;**
- **Definition| Modify;**
- **Definition| Create Undefined List.**

Gebriikt men de **OMT Rules Check** of **Expression Check** toch, dan wordt het diagram door FOOT symbolen ontsierd. Deze kan men weer weghalen door de commando's **OMT Data Flow| Clear Errors** of **OMT State| Clear Errors**.

2. Informatiesysteem en informatiekunde

2.1 Informatiesysteem

Om te begrijpen wat een informatiesysteem is moeten we eerst de begrippen informatie en informatietechnologie uitleggen.

Informatie is kennis in beweging tussen mensen, vastgelegd in de vorm van symboolstructuren. Over wat kennis is kunnen we lang discussiëren; filosofen zijn daar al tweeduizend jaar mee bezig. Dat doen we hier liever niet en daarom kiezen we een misschien niet heel precieze, maar voor ons doel wel bruikbare omschrijving: *kennis* bestaat uit beweringen die door hun mate van geldigheid door anderen te gebruiken zijn. *Symbolen* zijn tekens die krachtens een culturele conventie tot een bepaald type horen. Een symbool is bijvoorbeeld een letter op papier of een magnetisch streepje op een harddisk van een computer. Symbolen kunnen we aaneenrijgen tot structuren zoals bijvoorbeeld woorden en zinnen, waaraan we een betekenis toekennen. De wereld van de symbolen en symboolstructuren is de wereld van de taal. Om een beetje goed met elkaar te kunnen communiceren hebben we de signalen of symbolen van de taal nodig.

Informatietechnologie is de technologie op het gebied van de behandeling van informatie door automaten die zijn gerealiseerd met behulp van computers, programma's en technische communicatienetwerken (Gazendam, 1993: 17). Tegenwoordig wordt er ook veel over informatie- en communicatietechnologie gesproken. Deze naamgeving lijkt echter geen goede uitbreiding omdat zij een deelgebied van de hardware, namelijk de technische communicatienetwerken meer centraal stelt, terwijl het gaat om de programmeerbaarheid van automaten. We spreken toch ook niet over informatie- en beeldschermtechnologie terwijl het beeldscherm het meest in het oog lopende onderdeel van de tegenwoordige computer is. Ook geeft het begrip communicatie geen uitbreiding op het meer fundamentele begrip informatie, waarin communicatie besloten zit. *Informatie* is immers kennis in beweging tussen mensen. Het begrip 'informatie' veronderstelt de beweging van de communicatie evenals de symboolstructuren van de taal en het ontstaan en gebruik van menselijke kennis.

Een *informatiesysteem* is een systeem dat zelfstandig handelingen uitvoert gebaseerd op een programma en eigen waarnemingen, gerealiseerd met middelen uit de informatietechnologie, en in staat om menselijke taken op het gebied van besluitvorming, coördinatie, beheersing, kenniswerk en kantoorwerk in een organisatie te ondersteunen. In plaats van over een informatiesysteem kunnen we ook spreken over een *informaat*, dat wil zeggen een informatieverwerkende automaat (Dietz, 1992: 9). Dan wordt benadrukt dat een informatiesysteem een soort automaat is, een systeem dat zonder menselijke tussenkomst, gebaseerd op een programma en eigen waarnemingen, handelingen kan verrichten. Heel veel informaten zijn erg beperkt in hun waarnemingsvermogen en kunnen bijvoorbeeld alleen toetsaanslagen, muisgebruik en netwerkverkeer waarnemen. Door middel van barcodescanners en andere sensoren kan het waarnemingsvermogen van een informaat worden uitgebreid.

Een informatiesysteem omvat in de hierboven gegeven definitie geen mensen, in tegenstelling tot veel Nederlandse definities waarin mensen en procedures vaak ook onderdeel van een informatiesysteem kunnen zijn. Om het informatieverwerkend aspect van een

organisatie aan te duiden, waarin zowel mensen als informaten zitten, gebruiken we het woord *organisatorisch informatiesysteem*.

Informatiesystemen berusten in veel gevallen op de samenwerking van vele op computers, bijvoorbeeld PC's en servers, draaiende programma's die met elkaar zijn verbonden door middel van communicatienetwerken. Deze programma's moeten elkaar kunnen begrijpen op grond van een gemeenschappelijk taalsysteem, en moeten met elkaar moeten kunnen samenwerken door goed gedefinieerde bedrijfsprocessen, die op allerlei punten zijn gekoppeld aan de bredere, menselijke organisatie.

Een informatiesysteem is in de loop van de tijd geëvolueerd van een systeem dat een printer op een intelligente manier aanstuurt bij het maken van facturen, salarisspecificaties, betalingsopdrachten, e.d., via een interactief hulpmiddel voor het doen van taken door mensen (kantoorautomatisering; symbiotische relatie van werkende mens met informatiesysteem) naar een actor die zelfstandig beslissingen neemt over financiële stromen, materiaalstromen en documentstromen (logistiek, workflow).

Met deze verschillende verschijningsvormen corresponderen ook verschillende manieren van specificatie van het informatiesysteem: de documentstroom-methodegroep voor documentproductie, de actor-interactie-methode-groep voor het ondersteunen van interactieve taken en de bedrijfsproces-methode-groep voor het specificeren van de automatische besturing van bedrijfsprocessen.

2.2 *Informatievoorziening en informatiehuishouding, automatisering en informatisering*

Op het gebied van informatiesystemen zijn drie aggregatieniveau's te onderkennen: informatiesysteem, informatievoorziening en informatiehuishouding. De begrippen informatievoorziening en informatiehuishouding hebben betrekking op verzamelingen of stelsels van informatiesystemen. De *informatievoorziening* van een organisatie is het geheel van informatiesystemen waarmee in de behoefte aan informatie van die organisatie wordt voldaan (Structuurschetsen, 1983: 45). De *informatiehuishouding* van een stelsel van bestuurlijke eenheden is het geheel van informatiesystemen van die eenheden en de informatie-uitwisseling tussen die eenheden op grond van hun onderscheiden taken en bevoegdheden (Besturing Informatievoorziening O&W, 1985).

Automatisering is het proces waarbij handmatige arbeid geheel of gedeeltelijk door de werking van automaten wordt vervangen. Een *automaat* is een apparaat dat zonder menselijke tussenkomst een aantal elkaar opeenvolgende handelingen kan verrichten.

Informatisering is het proces waarbij kenniswerk wordt vervangen of van aard verandert door de invoering of uitbreiding van informatiesystemen. Informatisering is een veel ruimer begrip dan automatisering. Het omvat automatisering maar ook processen zoals ingrijpende veranderingen van bedrijfsprocessen en taken, cultuurveranderingen door het overgaan van papier op electronica, en de opening van een nieuw venster op de wereld door middel van Internet.

Informatisering houdt onder meer in het gebruiken van elektronische documenten, elektronisch postverkeer, virtuele werelden en door computerprogramma's gestuurde bedrijfsprocessen. Informatisering kan worden opgevat als het proces waarbij informatiesystemen met name de kenniswerker steeds meer routinetaken uit handen nemen en

nieuwe gereedschappen aanbieden. Hierdoor wordt het mogelijk nieuwe elementen in taken op te nemen en strengere eisen aan taakuitvoering te stellen (bijvoorbeeld in termen van termijnen voor beantwoording van vragen).

Informatisering leidt in de regel tot veranderingen in de relatie tussen bestuurlijke eenheden, de organisatiestructuur, de bedrijfsprocessen en de informatiehuishouding (zie ook Frissen, 1996). Informatisering is een bestuurlijk probleem getuige de rampen en bijna rampen die op dit gebied in de overheid hebben plaatsgevonden.

2.3 Het vakgebied bestuurlijke informatiekunde

Informatiekunde is de leer der informatiesystemen en der informatisering.

Bestuurlijke informatiekunde is de informatiekunde op bestuurlijk en administratief gebied, en op het gebied van kenniswerk.

Technische informatiekunde is de informatiekunde op technisch gebied.

Informatica is de discipline die vanuit de wiskunde informatiesystemen bestudeert (informatie-mathematica).

Het vakgebied bestuurlijke informatiekunde is in veel opzichten te beschouwen als een aan de organisatielunde verwante discipline. Zo heeft de informatiekunde een heel eigen kijk op organisaties (zie Hoofdstuk 3). De informatiekunde moet als brug kunnen dienen tussen de organisatielunde en de informatica, en houdt zich daarom bezig met de vraag hoe verschijnselen en begrippen uit de wereld van de organisatie kunnen worden beschreven op zo'n manier dat de informaticus op grond daarvan een informatiesysteem kan bouwen. Die beschrijving is in het algemeen meer formeel dan de natuurlijke taal, maar niet zo formeel als de wiskunde of de logica. Wel zijn veel beschrijvingsmethoden gebaseerd op een bepaald filosofisch, wiskundig of logisch systeem, bijvoorbeeld de semiotiek of de Petri-net-algebra.

Rond bestuurlijke informatiekunde bestaan twee misverstanden die uit de weg moeten worden geruimd: het zou gaan over computers en het zou gaan over informatie.

Het eerste misverstand is dat bestuurlijke informatiekunde gaat over *computers*. Het vakgebied bestuurlijke informatiekunde houdt zich bezig met informatiesystemen. Informatiesystemen zijn *artefacten*, dat wil zeggen constructies die door mensen gemaakt zijn die vergelijkbaar zijn met bijvoorbeeld romans.. Door de steeds grotere mogelijkheden van de informatietechnologie is er een steeds grotere vrijheid bij de vormgeving van dergelijke artefacten. Net als de literatuurwetenschap houdt de informatiekunde zich vooral bezig met de *fantasie* en de *vakkundigheid* die nodig is om een goed artefact te scheppen. Het gaat dus niet primair om de eigenschappen van de computer, evenmin als de literatuurwetenschap zich met de eigenschappen van pen en papier of de boekdrukkunst bezig houdt. Die eigenschappen zijn hoogstens als randvoorwaarde van belang. Wel houdt de informatiekunde zich evenals de taalkunde bezig met de gebruikte vormen en structuren. Echter, als die bestudering van vormen en structuren heel wiskundig gebeurt, is het geen informatiekunde meer maar informatica (informatie-mathematica).

Het tweede misverstand is dat bestuurlijke informatiekunde vooral gaat over *informatie*. Informatiekunde gaat niet in de eerste plaats over informatie, maar over *informatiesystemen*, systemen die informatie registreren, verwerken en opslaan. Informatiekunde houdt zich

vrijwel altijd in behoorlijk abstracte zin bezig met informatie, namelijk met de vormen en structuren waarin informatie zich voordoet. Wetenschapsgebieden die zich meer concreet met informatie bezig houden zijn bijvoorbeeld de management control (welke informatie heeft een controller nodig) en de documentaire informatiekunde (welke informatie heeft een wetenschapsbeoefenaar nodig en hoe kan de inhoud van documenten worden beschreven en geclassificeerd).

3. Organisatie en informatiesysteem

3.1 Typen informatiesystemen

Op het gebied van het typeren van informatiesystemen, informatievoorziening en informatiehuishouding zijn er indelingen die berusten op:

- vormgeving taakondersteuning (niveau: informatiesysteem);
- organisatorische vormgeving (niveau: informatievoorziening);
- taakverdeling tussen organisaties (niveau: informatiehuishouding).

De *vormgeving van de taakondersteuning* is de traditionele invalshoek voor het onderscheiden van typen *informatiesystemen*. Deze indeling berust op de volgende ideeën:

- een ideaal van integratie en samenhang
- een ideaal van hiërarchische planning en beheersing
- een ideaal van transactieverwerkende systemen als basis voor alle ‘hogere’ informatiesystemen

Laudon en Laudon (1996: 16) geven een bruikbare indeling:

- strategische informatiesystemen (EIS)
- managementinformatiesystemen (MIS)
- beslissingsondersteunende systemen (DSS)
- kenniswerksystemen (KWS)
- kantoorwerksystemen (OAS)
- transactieverwerkende systemen (TPS)

De *organisatorische vormgeving* is een indeling van typen *informatievoorziening* die berust op de volgende ideeën:

- hiërarchische planning en beheersing, integratie en samenhang zijn niet de enig mogelijke idealen; zij zijn in veel gevallen onwenselijk en onrealistisch;
- informatietechnologie verruimt de mogelijkheden om organisaties vorm te geven
- er is keuzevrijheid bij de vormgeving van organisaties;
- metaforen kunnen helpen bij de vormgeving van informatiesystemen en organisaties.

We gebruiken de volgende indeling (Gazendam, 1993: 282-293)

- informatiesysteem als fabriek (*mill*);
- informatiesysteem als organisme (*cell*);
- informatiesysteem als intelligente assistent (*mind*).

Een informatiesysteem als *fabriek (mill)* wordt gekenmerkt door een nadruk op de kwantiteit van de verwerkte informatie, die volgens vaste regels en patronen (stromen) moet worden verwerkt. Deze metafoor leidt in zijn meest uitgewerkte vorm tot een *procesorganisatie*, een organisatie waarin het informatiesysteem de logistieke besturing in handen heeft en veel mensen binnen de grenzen van deze sturing moet werken. Dit leidt tot een situatie waarin de mens als het ware ondergeschikt is aan het informatiesysteem. Deze metafoor kan goed worden toegepast op bijvoorbeeld de logistieke besturing van Albert Heijn waarbij een informatiesysteem de hoofdrol heeft en de schappenvullers binnen de besturingsgrenzen van dit informatiesysteem moeten werken.

Een informatiesysteem als *organisme (cell)* wordt gekenmerkt door een nadruk op de soepele, aangepaste omgang met mensen en de noodzaak tot symbiose van mens en informatiesysteem. Het informatiesysteem heeft het gebruik door de mens nodig om te

overleven; niet gebruikte (delen van) informatiesystemen kunnen worden gezien als afgestorven. Het informatiesysteem verkeert in een evolutionaire concurrentie met andere informatiesystemen (denk aan de concurrentie van Microsoft Word en Corel Wordperfect). Deze situatie leidt op de lange duur tot congruentie van informatiesysteem en organisatie (*congruentie-hypothese*). Er kan worden gezegd dat mens en informatiesysteem gelijkwaardig zijn. Deze organisme metafoor kan goed worden toegepast op veel kantoorwerkssystemen en kenniswerkssystemen.

Een informatiesysteem als *intelligente assistent (mind)* wordt gekenmerkt door een nadruk op vermogens als gebruik van kennis, zelfstandig optreden en lerend vermogen. Een dergelijk informatiesysteem wordt gezien als een gewillige ondergeschikte van de mens die op een relatief intelligente en zelfstandige manier taken uitvoert. Een dergelijke rol wordt wel met *agent* aangeduid. Zo zijn bijvoorbeeld op het Internet vele agents actief met informatie verzamelen voor hun menselijke meesters. Het gebruik van de intelligente assistent (mind) metafoor leidt tot een *netwerkorganisatie* waarin mensen met behulp van agents met elkaar samenwerken in locale en globale netwerken zoals Internet.

De *taakverdeling tussen organisaties* is een typering van de rol die een informatiesysteem op zich heeft genomen in maatschappelijk verband. Het is een indeling van typen informatiesystemen binnen een *informatiehuishouding*. Er worden de volgende taken op het gebied van het laten functioneren van een interorganisatorische informatiehuishouding onderscheiden:

- (1) beheren van regels (sectorsystemen, bijvoorbeeld onderwijs);
- (2) registreren van objecten (basisregistraties, bijvoorbeeld bevolking, vastgoed);
- (3) toepassing regels op objecten (systemen per organisatie-eenheid, bijvoorbeeld sociale dienst);
- (4) coördinatie op aspecten (aspectsystemen voor financiën, personeel en documenten).

De eerste twee taken behoren tot het informatiebeheer, de laatste twee taken tot het procesbeheer (zie Paragraaf 3.3. voor een uitleg van deze twee typen taken).

N.B. sectorsystemen bevatten veelal ook gegevens over objecten.

3.2 *Het gebruik van informatietechnologie*

De manier waarop informatiesystemen worden vormgegeven kan worden gezien als voortvloeiend uit het -in de tijd gezien- steeds veranderende spel van vraag en aanbod op de markt van informatietechnologie (Simons, 1989). In het afgelopen decennium is er aan de *aanbodkant* van de informatietechnologie een ontwikkeling te bespeuren in de richting van taakverdeling in de bedrijfskolom. Dit betekent dat bedrijven zich er op hebben toegelegd om concurrerend te zijn door een bepaald soort producten goedkoper en met hogere kwaliteit te leveren. De bedrijfskolom in de informatietechnologie ziet er (aan de hand van de soorten producten) ongeveer als volgt uit:

- besturingssystemen (Microsoft, IBM, UNIX, e.d.);
- programmeertalen, database-systemen. Middleware (Borland, Microsoft, IBM, Oracle, Ingres, e.d.);
- applicaties voor generieke taken zoals office pakketten (Microsoft, Corel, Lotus, e.d.);
- generieke applicaties voor specifieke taken in te stellen met bedrijfsspecifieke parameters (SAP, Triton, BAAN, e.d.);

- branchegerichte programmatuur en andere componenten (referentieobjectmodellen, architecturen, criteria) (Exact, Grote Beer, e.d.).

Eén en ander betekent dat er aan de *vraagkant* een ontwikkeling te zien is van zorgvuldig plannen van wat men zelf gaat uitvoeren (make) naar het tamelijk ad-hoc uitkiezen van de meest geschikte producten op de markt (buy). Met andere woorden, aan de vraagkant is er een ontwikkeling van make naar buy. Wat geschikt gekocht kan worden, wordt gekocht. Datgene wat overblijft om zelf te doen is veelal kleinschalig en moet dat ook zijn. Het voordeel van iets zelf doen, en niet uitbesteden moet vooral liggen in de benutting van specifieke *kennis* die in de organisatie aanwezig is. Dus *wel*: bedrijfsprocessen herontwerpen, parameters invullen en het gegevensbankontwerp uitbreiden, en *niet*: generieke componenten e.d. ontwikkelen. Projecten mogen niet langer duren dan drie à zes maanden als ze door informatiekundige specialisten worden uitgevoerd, en moeten binnen enkele weken klaar zijn als ze door z.g. eindgebruikers worden gedaan. Deze kleinschalige, snelle systeemontwikkeling staat bekend als rapid application development (RAD) of model-based application development (MAD) (Kranenburg en Van Riel, 1995).

Een organisatie kan op de volgende manieren tegenover nieuwe informatietechnologie staan:

<i>Houding</i>	<i>Kenmerken</i>
Innoverende houding:	<ol style="list-style-type: none"> 1. men is sterk innoverend 2. nieuwe producten worden vrij snel aangeschaft; 3. er wordt veel zelf gedaan en er wordt weinig uitbesteed 4. producten worden snel vervangen
Afwachtende houding:	<ol style="list-style-type: none"> 1. men neemt innovaties over als ze passen in het bedrijfsbeleid 2. men wacht af tot producten zich hebben bewezen en koopt ze dan in op de markt 3. er wordt weinig zelf gedaan en veel uitbesteed 4. producten worden na een normale economische levensduur (ca. 4 jaar) vervangen
Verzetshouding:	<ol style="list-style-type: none"> 1. men verzet zich tegen vernieuwingen op het gebied van informatietechnologie en houdt liever vast aan handmatige werkwijzen, bijvoorbeeld vanwege onvoldoende deskundig personeel of vanuit het oogpunt van bedrijfsstijl 2. men voert informatietechnologie of vernieuwingen in de informatietechnologie alleen in als het niet anders kan (bijvoorbeeld op het moment dat er echt geen voorraden voor de verouderde computer zijn te krijgen) 3. alles wordt uitbesteed 4. producten worden pas vervangen als ze (na ca 10 jaar of meer) technisch niet meer werken

3.3 *Verantwoordelijkheden op het gebied van informatiesystemen*

Procesbeheer betreft het uitvoeren van bedrijfsprocessen met behulp van informatiesystemen. Men is verantwoordelijk voor het op een juiste wijze laten verlopen van transacties en transformaties inclusief de verantwoordelijkheid voor het vastleggen van de informatie over die transacties en transformaties. Men spreekt ook wel van de verantwoordelijkheid voor het *gebruik* van een informatiesysteem.

Informatiebeheer betreft het beheren van de registraties over objecten zoals mensen, vastgoed, organisaties, voertuigen, en over de regels die in de uitvoering van taken worden toegepast. Het gaat hier om secundaire systeemfuncties die dienen om de primaire systeemfunctie van procesbeheer te kunnen realiseren. Behalve gegevens over objecten zijn altijd wettelijke regels of bedrijfsregels nodig bij het uitvoeren van taken. Men denke bijvoorbeeld aan de regels voor het toekennen van een bijstandsuitkering of aan de regels voor een effectief voorraadbeheer.

Systeembeheer is het beheer van de apparatuur, netwerken, programmatuur en bestanden (ANPB-voorzieningen) nodig om de primaire en secundaire systeemfuncties mogelijk te maken. Het maken van back-ups is bijvoorbeeld een onderdeel van het systeembeheer. Onder het systeembeheer vallen zowel het verwerven (kopen of maken) van de ANPB-voorzieningen als het onderhoud ervan en het ondersteunen van gebruikers

3.4 De functionaliteit van het informatiesysteem

Datgene wat een informatiesysteem moet doen noemen we de functionaliteit van het informatiesysteem. De functionaliteit bestaat uit een verzameling systeemfuncties. Elke systeemfunctie beschrijft dus een deel van datgene wat het informatiesysteem geacht wordt te doen. Systeemfuncties zijn er in soorten en maten. Twee belangrijke indelingen van systeemfuncties zijn een indeling naar het *type taak* dat wordt ondersteund en een indeling naar het *type ondersteuning* dat wordt gegeven.

De hoofdingeling van systeemfuncties naar het type taak dat wordt ondersteund volgt de indeling van verantwoordelijkheden:

- ondersteuning procesbeheer (primaire systeemfuncties);
- ondersteuning informatiebeheer (secundaire systeemfuncties);
- ondersteuning systeembeheer (tertiaire systeemfuncties).

Hoe de ondersteuning van taken gebeurt is sterk afhankelijk van de inrichting van de organisatie met behulp van het informatiesysteem in combinatie met de vormgeving van het artefact informatiesysteem zelf. Er zijn drie typen taakondersteuning:

- sterk interactieve ondersteuning van taken (bijvoorbeeld kantoorautomatiseringstaken);
- zelfstandige uitvoering van relatief grootschalige of complexe taken door het informatiesysteem onder supervisie van een mens of andere actor (bijvoorbeeld het maken van een verzameling betalingsopdrachten, het drukken van salarisspecificaties, het laten lopen van een simulatiemodel, of het uitwerken van een advies);
- permanente besturing van de organisatie door het informatiesysteem (regeling van informatie-, geld-, document- en materiaalstromen); in dit geval spreekt men van een *procesorganisatie*.

Deze indeling berust op de mate waarin door de mens besturingsimpulsen moeten worden gegeven om het informatiesysteem actief te doen zijn: van min of meer voortdurende impulsen tot een eenmalige impuls (aanzetten van het permanente besturingssysteem). In het laatste geval geeft het informatiesysteem aan mensen besturingsimpulsen.

3.5 Het gebruikersinterface van het informatiesysteem

Het gebruikersinterface van een informatiesysteem is datgene wat een gebruiker van dat informatiesysteem kan *zien*, en wat een gebruiker in concreto kan *doen* om het informatiesysteem aan te sturen. Van groot belang bij de bestudering van het gebruikersinterface is het nagaan van datgene wat een gebruiker moet *weten* om 'zien' en 'doen' op een soepele en effectieve manier te kunnen uitvoeren. Vooral over het doen gaan de principes van Schneiderman (1987). Het gebruik van een huisstijl gaat vooral over het zien.

Bij het ontwerpen van een gebruikersinterface moet er mee rekening worden gehouden dat er grofweg gezegd twee soorten gebruikers zijn: toetsgeoriënteerde gebruikers en muisgeoriënteerde gebruikers. Muisgeoriënteerde gebruikers kenmerken zich door een fijne motoriek en een visuele instelling. Zij kunnen reeksen aanslagen minder goed onthouden. Zij vinden het werken met een muis en met vensters fijn. Bij toetsgeoriënteerde gebruikers is soms een minder fijne motoriek aanwezig die leidt tot problemen bij langdurig muisgebruik (muishand of muisarm). Ook kan het voor sommige taken handiger zijn om met enige toetsaanslagen een resultaat te bereiken (een voorbeeld is de fietsenmaker die met vuile handen snel gegevens moet kunnen zoeken). Toetsgeoriënteerde gebruikers vinden het geen probleem om reeksen aanslagen te onthouden.

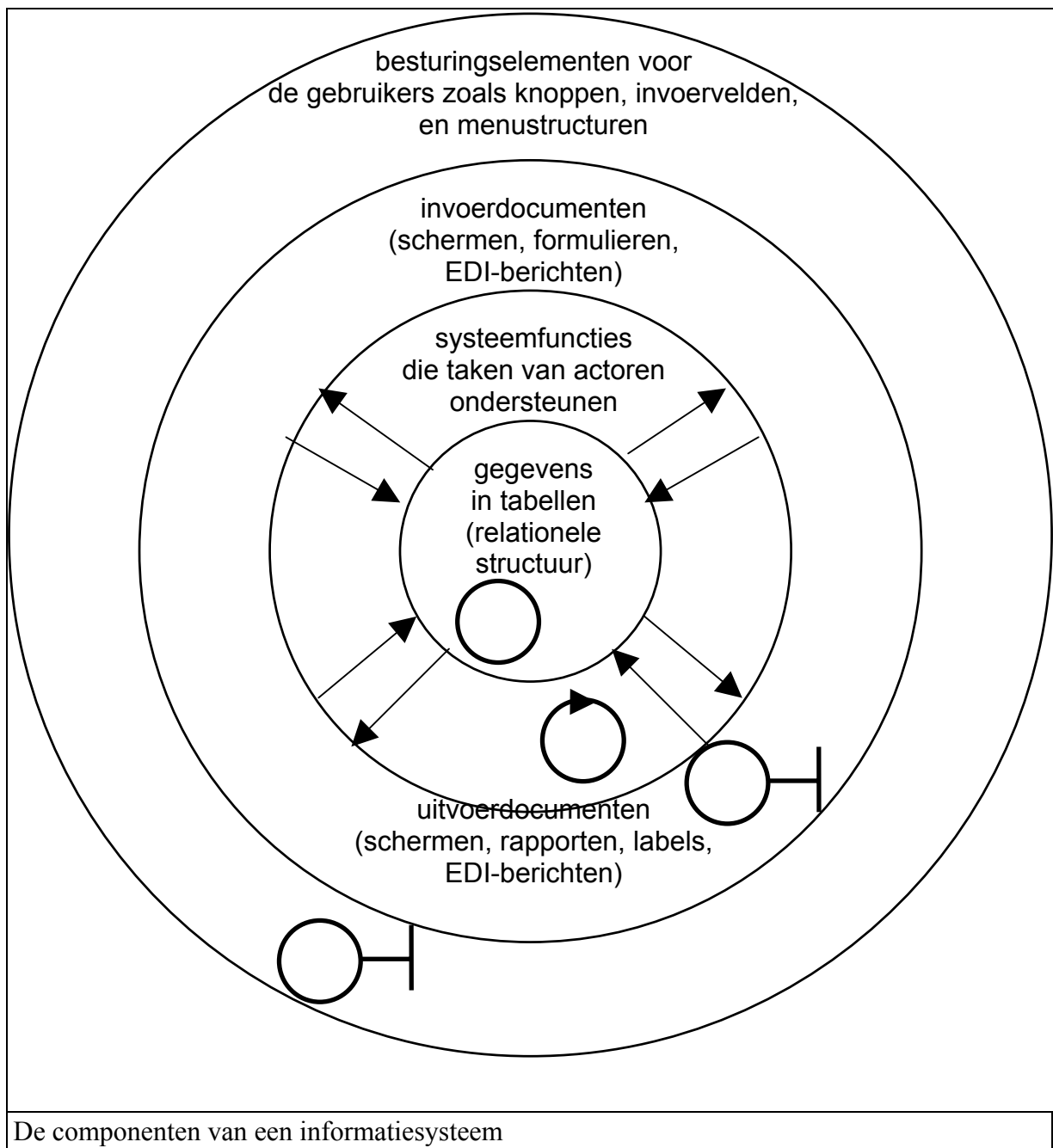
Een aantal principes voor het ontwerpen van een gebruikersinterface zijn:

1. (Zien)
zorg voor *consistentie* en begrijpelijkheid in taalgebruik, symbolen door het gebruiken algemeen aanvaarde conventies (bijvoorbeeld Windows, MS-Office)
2. (Zien)
zorg voor een heldere, onderscheidende *presentatie* die consistent wordt toegepast (huisstijl), onder meer op het gebied van eenheid van lettertype, kleurgebruik, lay-out;
3. (Zien)
ontwerp een *duidelijke werkomgeving* (duidelijke vensters en daarbij behorende stuulementen en dialogen) waarin het voor de gebruiker geen moeite kost om te zien wat er van hem/haar wordt verwacht en wat hij/zij kan doen;
4. (Doen)
geef de gebruiker het *initiatief* (laat de gebruiker aan het stuur zitten, niet de machine):
 - geef kortere wegen voor ervaren gebruikers;
 - geef een mogelijkheid om naar het vorige venster terug te keren;
 - geef altijd een 'undo' mogelijkheid;
5. (Doen)
geef duidelijke *feed-back* (vermijd 'dead air') en besteed aandacht aan de begrijpelijkheid en vriendelijkheid van de feed-back;
6. (Doen)
handel *fouten* goed af:
 - de gebruiker mag geen ernstige, rampzalige fouten kunnen maken; het systeem mag niet kunnen crashen;
 - geef eenvoudige en begrijpelijke foutmeldingen.
7. (Geheugen)
houd rekening met de *beperkingen aan het geheugen* van de gebruiker:
 - geen ingewikkelde combinaties van functietoetsen e.d.;
 - geef steeds aan wat de gebruiker kan doen (vermijd 'geheime' toetsaanslagen);

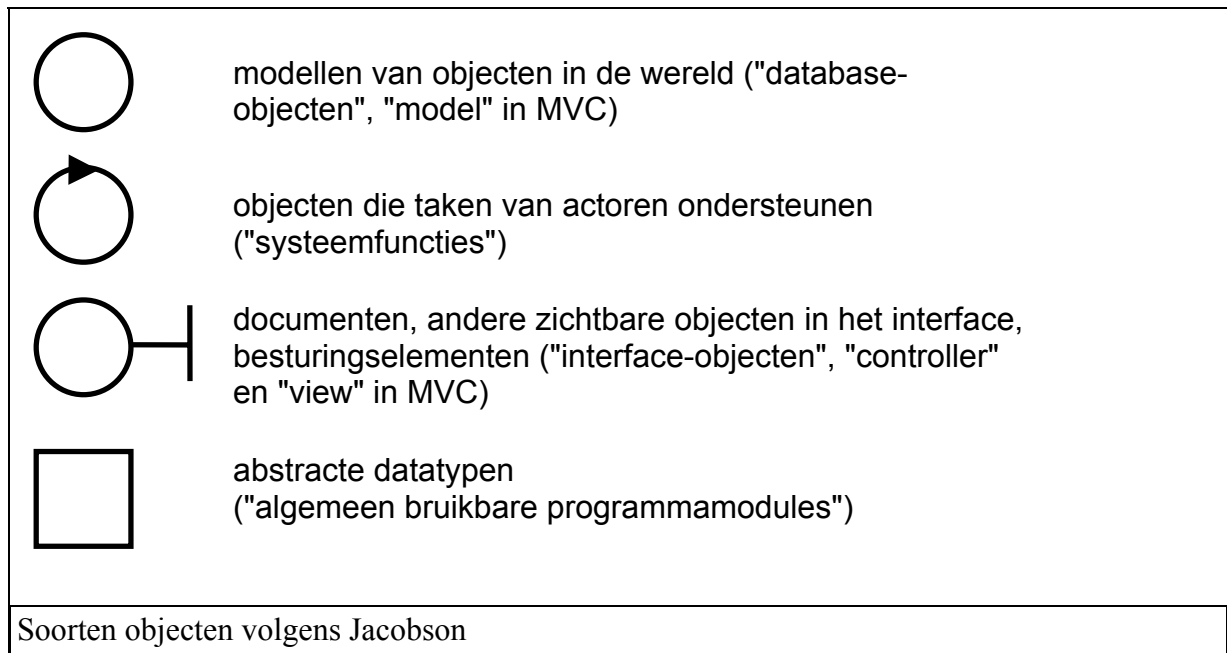
3.6 De opbouw van het informatiesysteem

Veel informatiesystemen kunnen worden gezien als opgebouwd uit de volgende subsystemen:

1. een schil van *documenten* die zijn gebaseerd op hiërarchische datastructuren (zoals formulieren) of netwerkdatastructuren (zoals HTML-pagina's);
2. een kern bestaande uit een *relationele gegevensbank*, en
3. een verzameling *taakgerichte modules* die taken uitvoeren en tussen de relationele gegevensbank en de schil van documenten gegevens transporteren en in een andere organisatievorm brengen;
4. *besturingselementen* voor de gebruiker zoals knoppen, invoervelden en menustructuren die ervoor dienen de documenten te wijzigen en de taakgerichte modules aan te sturen.



In bovenstaande figuur wordt deze architectuur van een informatiesysteem weergegeven. Hierin komen de volgende symbolen voor die soorten objecten volgens de notatie van Jacobson (1992) weergegeven.



Deze indeling van soorten objecten berust voor een deel op de Model-View-Controller (MVC) opvatting in het object-georiënteerd modelleren, waarbij de het modelleren van de wereld (model), het weergeven van dit model en andere objecten in het gebruikersinterface (view) en het besturen van het programma in reactie op de acties van de gebruiker (controller) als drie verschillende verantwoordelijkheden worden gezien.

4. Organisatie en interactie

4.1 Actor

Een *actor* is een systeem dat zelfstandig handelingen uitvoert gebruik makend van een stelsel van regels, een programma of script, adequaat reagerend op de omstandigheden, zodanig dat daarbij min of meer intelligente beslissingen worden genomen (Gazendam en Jorna, 1993; Carley en Prietula, 1994; Jorna, Gazendam, Heesen en Van Wezel, 1996: 20). Organisaties, mensen en in werking zijnde computers kunnen worden opgevat als actoren. Het beschouwen van mensen als actoren betekent overigens niet dat mensen niet meer creatief zouden mogen zijn of mogen improviseren of vage begrippen mogen gebruiken. Het tegendeel is het geval. Het is juist een uitdaging om deze verschijnselen te verklaren door een beroep te doen op goed definieerbare mechanismen in plaats van begrippen met een onduidelijke status. Ook betekent het hanteren van de actorbenadering niet dat machines gelijk of gelijkwaardig zijn aan mensen.

Volgens de *physical symbol system* theorie van Newell en Simon (1972) bestaat een actor (physical symbol system) uit een samenhangend geheel van sensoren, processoren en effectoren dat in staat is tot het zelfstandig toepassen en genereren van symboolstructuren zoals gegevens, kennis, en programma's (Gazendam, 1993: 14; Russell and Norvig, 1995: 31).

Bij actoren moeten we een onderscheid maken tussen actortype en individuele actor. Op grond van de gekozen beschouwingswijze kan elke individuele actor tot één of meer actortype worden gerekend. De actortypen worden onderscheiden op grond van de vervulde rol. Waar we in het navolgende bij de modellering over actoren spreken wordt steeds het *actortype* bedoeld. Daarnaast houden wij ons bij de modellering van organisaties veelal niet bezig met computer-actoren. In de praktijk van de modellering is een actor dus steeds een persoon of een groep van personen die een bepaalde rol vervult. Een persoon kan eventueel worden opgesplitst in de verschillende rollen die hij vervult en komt dan overeen met meerdere actoren. Dit is vooral bij kleine organisaties van belang.

Hoe onderscheiden we rollen? Hierbij is het onderscheiden van interacties (die hieronder worden uitgelegd) behulpzaam. Voor elke interactie is binnen het systeem één bepaalde actor verantwoordelijk. De rol van die actor is het besturen *of* het besturen en het uitvoeren van een bepaalde interactie.

4.2 Programma, impuls, actie, activiteit, proces

Rol, script en programma zijn (in toenemende mate meer formele) beschrijvingen van de handelingen die een actor onder bepaalde omstandigheden normaliter zal doen. Rol, script en programma zijn door een actor uitvoerbare beschrijvingen van handelingen. De meest formele beschrijving is een programma. Een *programma* is een samenhangende beschrijving van een hoeveelheid werk, inclusief eventueel de daarbij behorende doelen, eisen, randvoorwaarden, te volgen procedures en te volgen gedragsregels. Als de actor een computer is, dan is het programma een computerprogramma. Een *taak* is een programma waarvoor de verantwoordelijkheid aan een bepaalde actor is toegewezen. Om een actor een programma te laten uitvoeren is een gebeurtenis nodig die als *impuls* (trigger) werkt.

De uitvoering van een programma leidt tot feitelijk in de tijd plaatsvindende handelingen. Handelingen kunnen *acties* of *activiteiten* zijn. Om het verschil tussen actie en activiteit uit te leggen hebben we het systeembegrip nodig. Het objectstelsel kan worden beschreven in termen van *toestanden* en *gebeurtenissen* voor. Als een gebeurtenis optreedt hangt de volgende toestand zowel van de gebeurtenis als van de huidige toestand af (Rumbaugh e.a., 1991, p.89). Een toestand wordt in een toestandsdiagram (OMT module State) als afgeronde rechthoek getekend, een gebeurtenis als pijl.

Een *actie* is een handeling door een actor waarvan de tijdsduur te verwaarlozen valt in het kader van het reticulatieniveau van de toegepaste modellering. Een actie is geassocieerd met een gebeurtenis die als impuls werkt. Een *activiteit* is, in tegenstelling tot een actie, een handeling door een actor die tijd in beslag neemt. In veel gevallen is een activiteit geassocieerd met een toestand waarin het systeem bezig is met het uitvoeren van die activiteit.

Een *proces* is een verzameling samenhangende, door impulsrelaties verbonden, acties en activiteiten. Een proces dat op organisatieniveau wordt onderscheiden is een *bedrijfsproces*.

4.3 Organisatie

Een *organisatie* is een samenhangend geheel van (1) samenwerkende actoren, (2) door hen uitgevoerde processen van taakuitvoering en (3) door hen gegenereerde en gebruikte kennis en informatie (Gazendam, 1993: 14; Jorna. Gazendam, Heesen en Van Wezel, 1996: 20).

De grenzen van een organisatie kunnen worden vastgesteld op grond van kenmerken van de samenwerking. Een *werkorganisatie* is een verzameling samenwerkende actoren die wordt onderscheiden op grond van een stabiel patroon van samenwerkingsrelaties. Een organisatie kan ook worden afgebakend op grond van formele relaties tussen personen zoals eigendomsverhoudingen, arbeidscontracten, en dergelijke. In dat geval is een *formele organisatie* een verzameling samenwerkende actoren die wordt onderscheiden op grond van formele relaties tussen actoren. Die formele relaties zijn bijvoorbeeld eigendomsrelaties en contracten zoals arbeidsovereenkomsten. Werkorganisatie en formele organisatie hoeven niet congruent te zijn. Volgens Schmidt (1991) doet men er goed aan om de werkorganisatie als basis te zien en de formele organisatie te interpreteren als een extra laag van gedragsbepalende informatie.

Een organisatie bestaat niet alleen uit menselijke en machine actoren. Mensen en machines kunnen instromen en uitstromen in een organisatie, en toch blijft het karakter van een organisatie daarbij soms min of meer hetzelfde. Om deze reden is het van belang om naast actoren ook de in een organisatie plaatsvindende processen en de symboolstructuren (d.w.z. alle in documenten vastgelegde kennis en informatie) die bij een organisatie horen zoals de programma's van de bedrijfsprocessen, de gedragsregels, de briefhoofden, bij een organisatie te rekenen.

4.4 Zaak, prestatie

Een *zaak* is een voor menselijke beheersing vatbaar stoffelijk object (Algra, Ten Berge en Sleurink, 1986: B1/66).

Een *prestatie* is een handeling die voor degene waarvoor zij wordt verricht waarde heeft.

Prestaties kunnen in het samenspel tussen twee (of meer) organisaties worden geleverd (transactiegerichte prestaties) of binnen een organisatie (transformatiegerichte prestaties).

Een transactiegerichte prestatie kan zijn:

- het in eigendom overdragen van een zaak (P van product);
- het verlenen van een dienst (S van service);
- het verlenen of in eigendom overdragen van een recht (R van rights);
- het in eigendom overdragen van geld (F van finance).

Een *transactie* bestaat in de regel uit een transactiegerichte prestatie en een tegenprestatie. De *tegenprestatie* kan ook nil (N van null) zijn.

Transformatiegerichte prestaties bestaan in de regel uit:

- het creëren of van toestand veranderen van zaken (T van transformation)

Bij het veranderen van de toestand van een zaak kan men denken aan het veranderen van de tijdsdimensie (bewaren, opslaan), van plaats (transporteren) en van hoedanigheid (produceren).

4.5 *Subjectysysteem en objectysysteem*

Het te bestuderen probleemgebied beschouwen wij in termen van een subjectysysteem en een objectysysteem. Het *subjectysysteem* is het geheel van actief handelende actoren, gegroepeerd in systemen die we organisaties noemen. Het *objectysysteem* is het onderwerp van de handelingen der actoren. Het subjectysysteem kan onderdeel van het objectysysteem zijn, d.w.z. de handelende actoren kunnen zichzelf (mede) als onderwerp van handeling hebben (Dietz, 1992: 72).

Het subjectysysteem wordt bestudeerd met behulp van actor-interactiemodellen en bedrijfsprocesmodellen. Het objectysysteem wordt bestudeerd met behulp van documentmodellen en objectmodellen.

4.6 *Interactie, overdrachtsactie, communicatieve actie*

Een *interactie* is een geheel van acties en activiteiten door twee actoren dat een afgerond geheel vormt. Het hier gebruikte begrip 'interactie' komt overeen met het begrip 'transactie' uit de DEMO methode van Dietz (1996) en met het begrip 'use case' uit de methodes van Jacobson (1992) en Gale en Eldred (1996: 267). Een belangrijk verschil met deze methodes is dat wij uitgaan van begrippen uit het recht, de economie en de bedrijfskundige besturingstheorie voor het afbakenen van transacties en interacties. Transacties zijn in onze methode clusters van bij elkaar horende interacties die samen het economisch/bedrijfskundige begrip 'transactie' dekken. Interacties worden van elkaar onderscheiden op grond van de geleverde prestatie (een in oorsprong juridisch begrip); bij elke soort geleverde prestatie behoort een interactie.

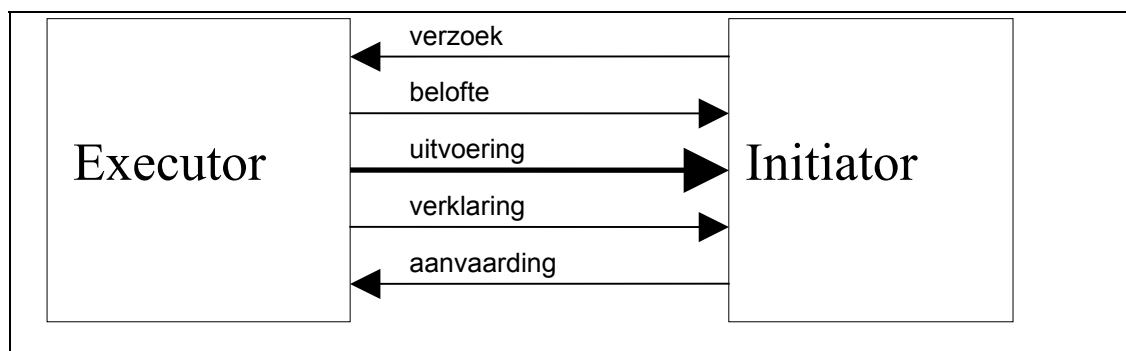
Bij essentiële handelingen zoals *overdrachtshandelingen* (overdrachtsacties) of *omzettingshandelingen* wordt veelal een prestatie geleverd. *Taalhandelingen* (communicatieve acties) bestaan uit op het verzenden en ontvangen van *berichten*. *Conversaties* bestaan uit het verzenden van berichten over en weer. Berichten of kopieën

daarvan kunnen worden bewaard in *berichtverzamelingen* die dienen als bron van informatie voor het handelen van actoren.

4.7 Algemene structuur van de interactie, actagene en factagene conversatie

De overdrachtsactie wordt uitgevoerd door de actor met de rol executor (E). De overdrachtsactie wordt door de executor ondernomen op initiatief van de andere actor, in de rol van initiator (I) (Dietz, 1996: 34). Voordat de overdrachtsactie wordt ondernomen, vindt eerst een actagene conversatie plaats waarbij het uitvoeren van de overdrachtsactie op de agenda van de executor wordt geplaatst. Nadat de overdrachtsactie heeft plaatsgevonden moet in een factagene conversatie worden geconstateerd dat zulks het geval is. De hieruit volgende algemene structuur van een interactie is volgens de DEMO methode (Dietz, 1996: 33):

- actagene conversatie, bestaande uit de communicatieve acties:
 - verzoek tot uitvoering van de overdrachtsactie door de initiator (OI);
 - belofte tot uitvoering van de overdrachtsactie door de executor (OE);
- uitvoering van de overdrachtsactie (eXecutie) (X) door de executor (EX);
- factagene conversatie, bestaande uit de communicatieve acties:
 - verklaring dat de overdrachtsactie is uitgevoerd door de executor (EE);
 - aanvaarding van de verklaring dat de overdrachtsactie is uitgevoerd door de initiator (RI) .



In een *actagene conversatie* komt een agendum tot stand, een punt op het actielijstje van de uitvoerder van de overdrachtsactie (Dietz, 1992: 78). De actagene conversatie bestaat uit een verzoek en een belofte:

- "Ik verzoek u mij 10 fietsen van het type Gazelle Tour de France Heren te leveren." (*verzoek* door JEEKAA, optredend als Initiator);
- "Ik beloof u deze fietsen binnen 12 weken te leveren onder voorwaarde van de bijgesloten leveringsvoorwaarden." (*belofte* door Gazelle, optredend als Executor).

In de *factagene conversatie* komt een feit tot stand, een door beide actoren geaccepteerde constatering (Dietz, 1992: 79).

- "Hierbij lever ik u 10 fietsen van het type Gazelle Tour de France Heren." (*verklaring* van Gazelle, optredend als Executor)
- "OK" (*aanvaarding* van deze verklaring door JEEKAA, optredend als Initiator)

Deze acties volgens het standaardpatroon van de interactie kunnen volgens de DEMO methode (Dietz, 1996: 33) als volgt worden ingedeeld in de volgende drie *interactiefasen*:

Opdrachtfase (O):

- verzoek (OI);
- belofte (OE);

Executiefase (E):

- uitvoering (EX);
- verklaring (EE);

Resultaatfase (R):

- aanvaarding (RI) .

Actagene conversatie en factagene conversatie vormen samen de besturing van de overdrachtsactie. De rol van de initiërende actor is daardoor besturend. De rol van de executerende actor is besturend én uitvoerend.

De hierboven genoemde stappen zijn de stappen die gedaan worden als de interactie op de standaard (default) manier verloopt. De uitzonderingen op dit standaardproces, zoals bijvoorbeeld het intrekken van het verzoek (bijvoorbeeld annulering van een bestelling) kunnen in een toestandsdiagram (zie Paragraaf 8.4) worden gemodelleerd.

4.8 *Transactie*

Een *transactie* is een geheel van interacties tussen actoren die tot verschillende organisaties behoren, waarbij op grond van een overeenkomst een uitwisseling van zaken of rechten plaatsvindt. Een transactie is een proces. De meeste gebruikelijke transactie bestaat uit een prestatie en een tegenprestatie, zoals het leveren van een fiets en het betalen van die fiets.

Deze definitie van 'transactie' komt overeen met het algemeen gangbare gebruik van het woord 'transactie'. In de economische transactiekostentheorie (TCE, transaction cost economics) wordt een onderscheid gemaakt tussen drie reticulatieniveau's waarop 'transactie' kan worden gedefinieerd (Williamson, 1985: 1):

- de transactie als overdracht van goed of dienst;
- de transactie als ruilproces waarin de fasen contact, contract en controle worden onderscheiden;
- de transactie als gebeurtenis waarbij rechten van eigendom of beheer worden overgedragen, te onderscheiden van het leveren van de goederen of diensten en het eventueel betalen van die goederen of diensten.

Het meest gedetailleerde reticulatieniveau, namelijk de overdracht van eigendom of beheer, komt overeen met ons interactie-beschrijvingsniveau. Ook het leveren van goederen of diensten is een interactie, evenals het betalen van deze goederen of diensten. Het meest globale reticulatieniveau van de TCE komt overeen met ons transactie-beschrijvingsniveau. Het ertussen liggende reticulatieniveau geeft ons de aanwijzing om naast contract (het afsluiten van een overeenkomst) en controle (hiertoe zullen we wel de interacties leveren en betalen moeten rekenen) ook een contact-interactie te onderscheiden. We krijgen zo een meer uitgebreiden transactiestructuur zoals die bijvoorbeeld door Gale en Eldred (1996: 282) is uitgewerkt:

- opnemen contact (een interactie waarin alleen informatie wordt uitgewisseld);
- afsluiten overeenkomst (als gevolg van onderhandelen) (R);
- leveren prestatie (meestal: leveren product of dienst) (P/S);

- leveren tegenprestatie (meestal: betalen) (P/S/F);
- verlenen/ verkrijgen nazorg (P/S).

In de overheid is het gebruikelijk om de volgende fasen te onderscheiden:

- raming van een uitgave (bijvoorbeeld de jaarlijkse rijksbijdrage aan een universiteit);
- autorisatie om deze uitgave te doen (b.v. doordat een begroting is goedgekeurd);
- verplichting (men heeft een bestelling gedaan of een subsidie toegekend);
- prestatie (er is geleverd of aan andere voorwaarden voldaan en derhalve een recht op betaling ontstaan);
- betaling.

Deze fasen kan men zien als fasen in een transactieproces. Bij dat transactieproces zijn dan meestal drie partijen betrokken, te weten de autoriserende instantie (bijvoorbeeld de Staten-Generaal), de uitvoerende instantie (bijvoorbeeld een minister) en een derde instantie (bijvoorbeeld een universiteit).

4.9 Transformatie

Een transformatie is een geheel van interacties tussen actoren die tot dezelfde organisatie behoren, waarbij zaken worden gecreëerd of van toestand worden veranderd. Waar bij een transactie de centrale plaats wordt ingenomen door overdrachtshandelingen, wordt binnen de transformatie de centrale plaats ingenomen door omzettingshandelingen (omzettingssacties). Een transformatie is een proces. De interacties van een transformatie kunnen eventueel ook zijn tussen een actor en zichzelf. We hebben dan te maken met een zelfactiverend en zelfsturend proces. In het geval van een zelfactiverend, zelfsturend proces kan aan de stappen in de standaard-structuur van de interactie de volgende betekenis worden toegekend:

Opdrachtfase:

- maken van een plan (verzoek);
- zich vastleggen om het plan uit te voeren (belofte);

Executiefase:

- uitvoeren van de geplande omzettingssactie (uitvoering);
- constateren dat de geplande actie is uitgevoerd (verklaring);

Resultaاتفase:

- beoordelen of de uitgevoerde omzettingssactie aan de gestelde eisen voldoet (aanvaarding).

5. Objecttype en associatietype

5.1 Object, begrip, symbool

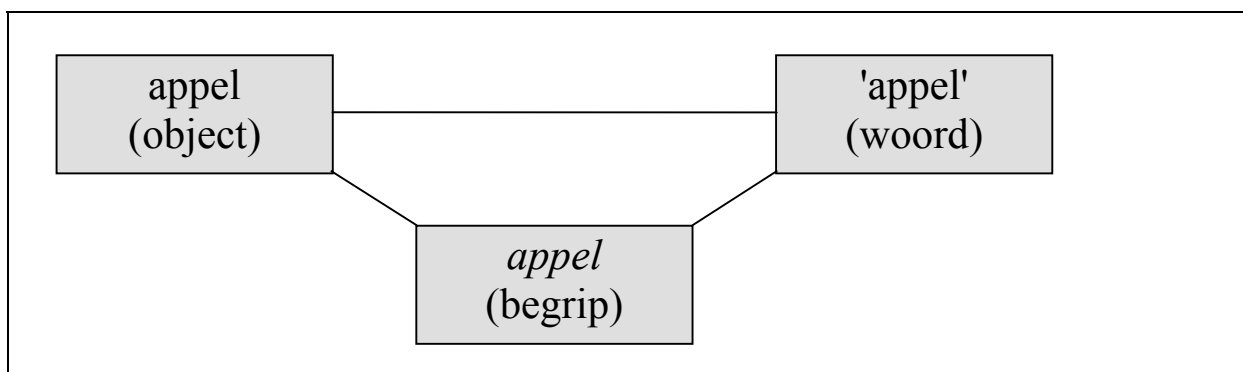
Een *object* is iets waaraan we een invariante identiteit toekennen. Een object kan worden aangewezen in de reële wereld, of door de menselijke geest worden geconstrueerd in de conceptuele wereld. Met andere woorden, een object is iets dat een afzonderlijk bestaan heeft in de reële of conceptuele wereld en daardoor met een (zo mogelijk) unieke naam wordt benoemd. Voorbeelden: die appel daar, Jan Jansen, Nederland, mijn banksaldo, de RUG.

Een *begrip* of *concept* is een iets dat door de menselijke geest is geconstrueerd en tot de conceptuele wereld behoort. Een concept heeft eigenschappen die al dan niet toepasbaar zijn op objecten. Voorbeelden van concepten zijn: het begrip *appel*, organisatie, democratie. Het begrip *appel* is toepasbaar op die appel daar. Democratie is toepasbaar op Nederland. Organisatie is toepasbaar op de RUG.

Een *symbool* is een inscriptie in een fysiek medium behorende tot een symbooltype (Gazendam, 1993: 64). Een *symboolstructuur* is een bij elkaar horende groep symbolen. De meest voorkomende symboolstructuren zijn woorden, zinnen en verhalen. Voorbeeld: het woord 'appel'. Ter onderscheiding van object, begrip en woord wordt het object appel gewoon geschreven, het begrip *appel* cursief of met hoofdletters (APPEL) en het woord 'appel' tussen aanhalingstekens.

Objecten, begrippen, symbolen en symboolstructuren zijn allen entiteiten. Een *entiteit* is iets waarover gesproken kan worden.

In de *semiotiek* (Peirce, 1958) wordt de drie-eenheid 'object - woord (of meer algemeen: symboolstructuur) - begrip' onderscheiden. Het object is datgene waarnaar wordt verwezen, het woord is datgene wat verwijst. Object, woord en begrip worden in een proces dat semiosis heet met elkaar verbonden. Het proces van semiosis verbindt de uitingshandeling met de propositionele handeling en de illocutionaire handeling. Dit kan het meest duidelijk worden uitgelegd voor het verband tussen het uiten van een woord (uitingshandeling) en het leggen van een verband met een object in de wereld en het toekennen van eigenschappen aan dat object (propositionele handeling). Het begrip dient als verzameling eigenschappen die sturend



is bij het proces van semiosis, en daar helpt bij het selecteren van het object en het koppelen van betekenis aan het woord. Bijvoorbeeld: het begrip *appel* dient om de appels te onderscheiden van de peren in een mand en het woord 'appel' daarvoor te kunnen gebruiken.

Informatiesystemen bevatten symbolen of symboolstructuren die naar objecten verwijzen. Om bruikbaar te zijn moet een informatiesysteem, conform de semiotiek, ook op de een of andere manier *begrippen* bevatten. De voor een informatiesysteem noodzakelijke begrippen behoren tot onder meer de *objecttypen*, *attribuuttypen*, *associatietypen* en *generalisaties*. De begripsstructuur waarop een informatiesysteem berust maakt het mogelijk dat de symboolstructuren in het informatiesysteem iets zeggen over de objecten in de fysieke of conceptuele wereld. Men zou zelfs kunnen zeggen dat informatiesystemen niet alleen afbeelden maar ook virtuele werelden creëren. Dat kan alleen maar dankzij het begrippennetwerk waarop een informatiesysteem berust.

5.2 Objecttype en attribuuttype

Een *attribuut* van een entiteit A is een entiteit B die de betreffende entiteit A beschrijft in termen van een waarde behorend tot een bepaald domein. Bijvoorbeeld het begrip *rood* beschrijft de appel die op mijn bureau ligt op het gebied (domein) van de kleur. Een *eigenschap* is de aanwezigheid van een attribuut met waarden binnen een bepaald bereik, of de aanwezigheid van een regel die het gedrag van een entiteit in bepaalde omstandigheden definieert. Bijvoorbeeld: deze appel is eetbaar.

Een *type* is een verzameling bij elkaar horende eigenschappen (Ter Bekke, 1988: 75; Gazendam, 1993: 92). Een type is een begrip dat zelf weer eigenschappen heeft. De belangrijkste eigenschappen van een type zijn:

- (1) de eigenschappen die bij het type behoren (de intensie van het type), en
- (2) de verzameling entiteiten die voldoen aan deze eigenschappen (de extensie van het type).

Een type is dus niet gelijk aan de verzameling entiteiten die van dat type zijn (Ter Bekke, 1988: 75). Deze verzameling is echter wel een eigenschap van het type.

Voorbeeld: de verzameling personen die medewerker is van de Faculteit Bedrijfskunde is niet gelijk aan het type MEDEWERKER FACULTEIT BEDRIJFSKUNDE. Het type kan min of meer constant zijn, terwijl de verzameling personen een steeds veranderende verzameling is.

De namen van types schrijven we met hoofdletters. Voorbeeld: Het type APPEL is gedefinieerd door de volgende eigenschappen: het is een vrucht van de boom behorende tot de familie der Pomaceae.

En *instantie* van een type is een entiteit die van dat type is. De **Appel op mijn bureau** is een instantie van het type APPEL. De namen van instanties schrijven we ter onderscheiding van de rest van de tekst **vet** en, eventueel behalve de beginletter, met kleine letters.

Een *objecttype* is een type dat van toepassing is op objecten. Een *attribuuttype* is een type dat van toepassing is op attributen.

5.3 Associatie, associatietype, generalisatie

Een *associatie* is een relatie tussen twee of meer objecten, bijvoorbeeld een deel-geheel relatie of een groep-groepslid relatie. De mogelijkheid van een bepaald object om een bepaald soort relatie te hebben wordt tot de eigenschappen van het object gerekend. Een associatie a_{PQ} tussen de instanties van de objecttypen P en Q kan worden beschreven als een verzameling

bestaande uit twee andere verzamelingen, namelijk de verzameling i_P instanties van objecttype P en de verzameling i_Q van het objecttype Q:

$$a_{PQ} = \{i_P, i_Q\}.$$

In de associatie a_{PQ} tussen instanties van de objecttypen P en Q is de cardinaliteit van de verzameling instanties van P n_P en de cardinaliteit van de verzameling instanties van Q n_Q . De verhouding tussen deze cardinaliteiten m_{PQ} noemen we de multipliciteit van de associatie.

$$m_{PQ} = \{n_P, n_Q\}. \text{ Dit wordt ook wel geschreven als: } m_{PQ} = n_P : n_Q$$

Voorbeeld: **Kamer 923**, behorende tot het objecttype KAMER, is een deel van het **Gebouw Landleven 5**, behorende tot het objecttype GEBOUW. **Kamer 923** is een instantie van KAMER. **Gebouw Landleven 5** is een instantie van GEBOUW. Ook **Kamer 949** is een deel van het **Gebouw Landleven 5**. Bij het **Gebouw Landleven 5** horen meerdere kamers als deel. De associatie $a_{\text{GEBOUW-KAMER}}$ tussen **Gebouw Landleven 5** en de genoemde kamers kan wiskundig worden beschreven als:

$$a_{\text{GEBOUW-KAMER}} = \{\{\text{Gebouw Landleven 5}\}, \{\text{Kamer 923}, \text{Kamer 949}\}\}.$$

De multipliciteit van deze associatie is:

$$m_{\text{GEBOUW-KAMER}} = 1 : 2.$$

Associaties behoren tot een *associatietype*. De namen of afkortingen van associatietype worden net als objecttypen met hoofdletters geschreven. De belangrijkste kenmerken van een associatietype zijn:

- (1) de *verzameling associaties* die ertoe behoort.
- (2) de *objecttypen* waartoe de deelnemende objecten behoren;
- (3) het *soort relatie* tussen de deelnemende objecten in een associatie;
- (3) de *multipliciteitsregel* van het associatietype, een regel voor de minimale en maximale cardinaliteit die elk van de in de associatie deelnemende deelverzamelingen van objecten mag hebben

Voorbeeld: de associatie $a_{\text{GEBOUW-KAMER}}$ behoort tot het associatietype $A_{\text{GEBOUW-KAMER}}$, ook wel geschreven als GEBOUW-KAMER. Dit associatietype heeft betrekking op de objecttypen GEBOUW en KAMER; de soort relatie is GEBOUW BEVAT KAMER. De multipliciteitsregel is dat één gebouw minstens één en maximaal meerdere kamers kan bevatten, en dat een kamer altijd tot één gebouw behoort. In termen van minimale en maximale cardinaliteit van de instanties van de deelnemende objecttypen betekent dit het volgende (zie tabel).

Associatie	Objecttype		Objecttype	
GEBOUW-KAMER	GEBOUW		KAMER	
	minimale cardinaliteit	maximale cardinaliteit	minimale cardinaliteit	maximale cardinaliteit
	1	1	1	M

Hierbij betekent M: meer, d.w.z. een geheel getal groter of gelijk aan twee.

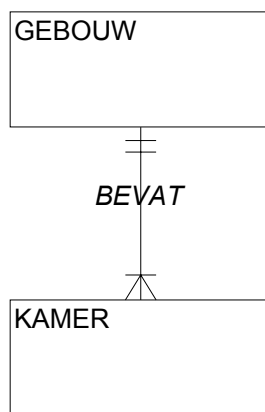
We spreken kortheidshalve van een 1:M associatietype als de maximale cardinaliteiten in de bijbehorende multipliciteitsregel 1 resp. M zijn. Op dit gebied zijn er de volgende mogelijkheden:

- 1:1 associatietype (één op één) of
- 1:M associatietype (één op meer) of
- M:M associatietype (meer op meer).

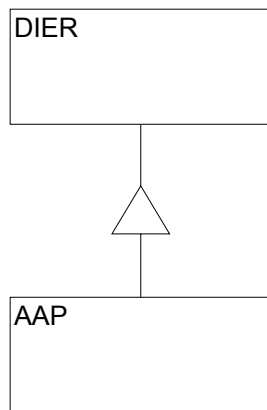
In objectmodellen wordt een objecttype door een rechthoek (met erin de naam van het objecttype in hoofdletters) weergegeven; een associatietype wordt door een lijn tussen de rechthoeken van de deelnemende objecttypen getekend. De multipliciteitsregel (in termen van minimale en maximale cardinaliteiten) wordt op de lijn weergegeven door de volgende symbolen:

- 0: een rondje (lijkt op een nul);
- 1: een streepje (lijkt op een één);
- M: een kraaietootje (lijkt op een M).

Voor ons voorbeeld van het associatietype GEBOUW-KAMER en de objecttypen GEBOUW en KAMER komt het objectmodel er als volgt uit te zien.



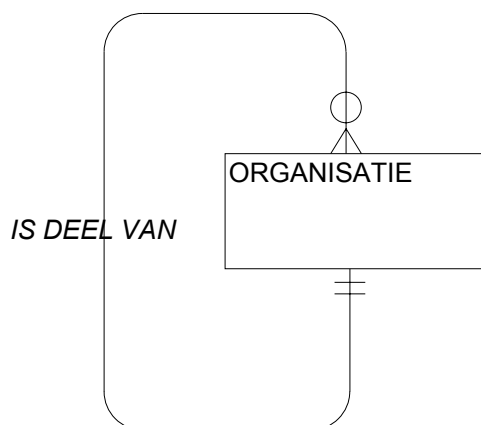
Een relatie soort die direct tussen de objecttypen zelf zit is de *generalisatie*. In een generalisatie-relatie tussen twee objecttypen zijn twee rollen: het supertype en het subtype. Het subtype heeft altijd tenminste alle definiërende eigenschappen van het supertype. Het supertype heeft daardoor altijd tenminste alle instanties van het subtype zelf als instantie. Voorbeeld: de *aap* is een subtype van het *dier*. De *aap* heeft daardoor alle eigenschappen van het *dier*. De verzameling instanties van het *dier* omvat in ieder geval alle apen. In het objectmodel ziet dit er als volgt uit.



5.4 Structuren

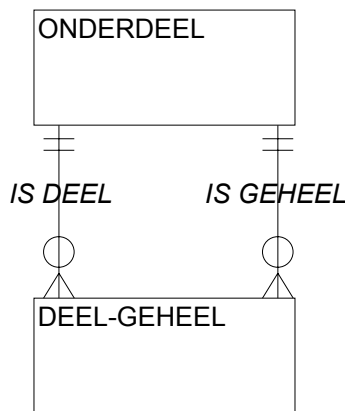
Een *structuur* bestaat uit een associatietype en de daarin deelnemende objecttypen. Voorbeelden zijn de organisatiestructuur, de 'bill of materials' structuur of de verpakkingsstructuur.

Een *organisatiestructuur* is meestal een hiërarchische *deel-geheel structuur*. Bijvoorbeeld: een universiteit bestaat uit een aantal faculteiten; elke faculteit bestaat uit een aantal vakgroepen; elke vakgroep bestaat uit een aantal personen. Er is dan een 1:M associatietype tussen organisatie en deelorganisatie. Het bijzondere van een organisatiestructuur is dat zij kan worden weergegeven als één objecttype ORGANISATIE dat tweemaal deelneemt in een associatietype (zie figuur). Op deze manier kan een hiërarchie van onbeperkte diepte worden weergegeven.



Sommige organisatiestructuren zitten ingewikkelder in elkaar, bijvoorbeeld de matrixorganisatie. Hierbij kan bijvoorbeeld een persoon van meerdere vak- of werkgroepen deel uitmaken. Er is dan sprake van een M:M-associatietype. Dit M:M-associatietype komt voor in de bill-of-materials structuur. Een bill of materials is een lijst waarop staat welke onderdelen moeten worden gebruikt in de assemblage van een groter onderdeel of eindproduct. Een *bill-of-materials structuur* is een M:M *deel-geheel structuur*, waarbij onderdelen deel kunnen uitmaken van meerdere grotere onderdelen. Een voedingseenheid kan

bijvoorbeeld deel uitmaken van zowel een printer als een computer. Een schroefje kan deel uitmaken van vele andere onderdelen. Deze M:M deel-geheel structuur ziet er in een objectmodel als volgt uit.



De bill-of-materials structuur is een productstructuur. Daarnaast kennen we ook een *verpakkingsstructuur*: een product, bijvoorbeeld een kaars, gaat in een doosje samen met nog een aantal andere kaarsen; dat doosje gaat in een kist; die kist wordt op een pallet geplaatst, het pallet gaat in een container; de container gaat in een schip. Eén en ander betekent dat een object van meerdere structuren deel kan uitmaken.

Deel-geheel structuren zijn in principe *recursieve structuren*, hetgeen wil zeggen dat een deel van het geheel zelf weer een geheel kan zijn dat uit delen bestaat, enzovoorts.

Deel geheel structuren zijn de belangrijkste structuren waarmee we te maken kunnen krijgen. Daarnaast bestaan er nog enkele andere typen structuren. De volgende typen *structuren* kunnen worden onderscheiden:

- geheel/ deel structuur(object A is een deel van object B);
- generalisatiestructuur (type A is een subtype van B).
- processtructuur (actie A veroorzaakt activiteit B; activiteit B veroorzaakt actie C).

5.5 Categorieën

Voor het ordenen en onderscheiden van objecttypen is het handig om over een lijst met categorieën te beschikken. Een *categorie* is een type objecttype. In deze paragraaf stellen wij ons tot doel om een handzame (d.w.z. niet uitputtende) lijst met categorieën te maken die van nut is bij het categoriseren van objecttypen in een objectmodel. Deze lijst moet aansluiten bij de gevolgde werkwijze van het modelleren van organisaties in termen van actoren en interacties.

De lijst met categorieën bouwen we als volgt op. We beginnen met het onderscheiden van *actoren*, *transacties*, en de *onderwerpen* van die transacties. Het onderwerp van een transactie is in de regel de *toestandsverandering* van meerdere *verzamelingen objecten*.

Transacties behoren tot de meer algemene categorie van de gebeurtenissen of processen (reeksen van gebeurtenissen). Een transactie is iets dat gebeurt, iets waarbij twee of meer actoren actief betrokken zijn. Transacties hebben altijd plaats op een bepaald tijdstip

(eventueel een tijdsperiode) en meestal ook op een bepaalde plaats. Transacties gaan ergens over, namelijk het onderwerp van de transactie.

Bij de transactie **Inkoop #26** gaat het over:

- drie fietsen van type Gazelle Superieur Special, 5 versnellingen, 57 hoog, heren, Boston Groen (verzameling **Inkoop #26 regel 1**);
- één fiets van het type Gazelle Paris, 57 hoog, dames, Daytona Violet (verzameling **Inkoop #26 regel 2**).

De verzamelingen die onderwerp zijn van **Inkoop #26** ondergaan in ieder geval de volgende toestandsveranderingen:

- zij veranderen van eigenaar;
- zij veranderen van plaats;
- zij worden betaald.

De toestandsverandering kan meer of minder zijn gespecificeerd, bijvoorbeeld alleen ‘in eigendom overdragen’ of daarnaast ook ‘bij JEEKAA te Schiermonnikoog op 5 december 1997’. De verzameling objecten kan *abstract* zijn: van de verzameling objecten is dan alleen de hoeveelheid en het type bekend, of *concreet*: elk van de objecten in de verzameling is dan identificeerbaar met een naam of nummer.

Verzamelingen moeten worden onderscheiden van *individuen* en *typen*. Een verzameling is bijvoorbeeld een aantal fietsen dat wordt besteld bij de leverancier. De fietsen in die verzameling kunnen abstract zijn. Een *individu* is bijvoorbeeld één bepaalde fiets met een bepaald framenummer. Een *individu* is altijd concreet. Een *type* is bijvoorbeeld een model fiets. We hebben hier dus de categorie *type*, die scherp moet worden onderscheiden van het begrip *objecttype*. Tot de categorie *type* behoort bijvoorbeeld het objecttype FIETSTYPE met instanties als **Gazelle Superieur Special, 5 versnellingen, 57 hoog, heren, Boston Groen**.

De verzamelingen worden meestal op grond van een bepaald doel onderscheiden, bijvoorbeeld het bestellen van fietsen. Bij *typen* geldt de regel dat elke verandering van een *type* in principe tot de definitie van een nieuw *type* leidt. Het fietstype Gazelle Tour de France heren wit 1994 is een ander *type* dan Gazelle Tour de France heren wit 1995 als bijvoorbeeld de prijs verschilt of als er belangrijke onderdelen verschillen. Typen blijven geregistreerd zolang ze relevant zijn, d.w.z. zolang er verzamelingen of individuen van dat *type* in de gegevensbank zitten of kunnen worden verwacht.

Meestal zijn *tijd* en *plaats* attributen van objecten. In sommige gevallen is het echter wenselijk om deze attributen tot objecttypen te verheffen (dit proces heet verdinglijking of reïficatie). Zo kan het in de logistiek wenselijk zijn om objectverzamelingen te onderscheiden op grond van de plaats waar zich objecten bevinden. Men kan dan bijvoorbeeld nagaan of de capaciteit van magazijn C (maximum capaciteit 250 items) is benut. In dat geval komt er een objecttype *plaats* en een objecttype voor de verzameling objecten die zich op die plaats bevinden. Bij het roosteren kan daarnaast het verdinglijken van *tijd* een rol spelen. Men moet bijvoorbeeld na kunnen gaan of in week 32 de nachtdienst is ingeroosterd.

Eén en ander leidt tot de volgende categorieën:

1. actor;
2. transactie (meer algemeen: gebeurtenissen en processen);
3. individu;
4. verzameling;

5. type;
6. plaats;
7. tijd.

In de regel zullen we de categorieën 1 t/m 5 gebruiken voor het ordenen en onderscheiden van objecttypen.

6. Verbale beschrijving van organisatie en informatiesysteem¹

6.1 Overzicht van te beschrijven aspecten van organisatie en informatiesysteem

Onze bestuurlijk-informatiekundige modellering van een organisatie begint met het verbaal beschrijven van die organisatie en informatiesystemen.

Deze beschrijving bestaat uit drie gedeelten:

1. een beschrijving van hoe de organisatie met informatiesystemen omgaat;
2. een beschrijving die later moet worden gebruikt bij het maken van modellen;
3. een beschrijving van één der informatiesystemen in de organisatie.

De onderdelen van de verbale beschrijving worden hieronder in een tabel weergegeven.

<i>1. Organisatie en informatiesystemen</i>	
Typen informatiesystemen	
Houding van de organisatie tegenover informatietechnologie	
De verantwoordelijkheidsverdeling op het gebied van informatiesystemen	
<i>2. Gegevens nodig voor het maken van modellen</i>	
<i>Aspect</i>	<i>Van belang voor</i>
Typologie Starreveld	beoordeling informatiegebruik bij D-model
Organigram	actoren in B1-model
Bedrijfsprocessen	B2-model
Gegevensstromen en bestanden	D-model
<i>3. Beschrijving van een informatiesysteem</i>	
De functionaliteit van het informatiesysteem	
Het gebruikersinterface van het informatiesysteem	
De opbouw van het informatiesysteem	

Het is bij het beschrijven van een bepaald onderdeel van belang om steeds drie stappen te doen:

- het verzamelen van voldoende feiten;
- het op een zorgvuldige manier beredeneren waarom een bepaald kenmerk aan een organisatie of informatiesysteem wordt toegekend (het interpreteren van de feiten);
- het toekennen van het kenmerk.

Zowel de feiten als de beredeneerde interpretatie moeten waar mogelijk worden weergegeven. Alleen het toekennen van een kenmerk zal in de meeste gevallen onvoldoende zijn.

6.2 Typen informatiesystemen

Het te beschrijven informatiesysteem wordt getypeerd op drie manieren:

¹Vermeld steeds de bronnen (documenten, interviews, etc.) op grond waarvan u tot bepaalde constatering of conclusies komt.

- hoe het vorm geeft aan taakondersteuning (niveau: informatiesysteem);
- welke invloed het heeft op de organisatorische vormgeving (niveau: informatievoorziening);
- welke rol het speelt in de taakverdeling tussen organisaties (niveau: informatiehuishouding).

Voorbeeld

Het informatiesysteem XP van de Sociale Dienst te Z. heeft de volgende kenmerken:

1. Het systeem is een combinatie van een transactieverwerkend systeem met een beslissingsondersteunend systeem en een expertstelsel (dit is een soort kenniswerksysteem (KWS)). Het transactieverwerkend systeem (TPS) zorgt voor de betaling van uitkeringen volgens de geldende regels. Het expertstelsel helpt bij het invoeren van gegevens over nieuwe cliënten. Het beslissingsondersteunend systeem bevat modellen en scenario's om beleid op te baseren.
2. Het systeem versterkt de bureaucratie in de organisatie. Het is een type mill metafoor systeem.
3. Het systeem heeft de rol van toepassing van regels op objecten. Het bewaart ook gegevens die over objecten zijn geregistreerd; deze gegevens zijn echter gerelateerd aan de gemeentelijke bevolkingsadministratie.

6.3 *Houding van de organisatie tegenover informatietechnologie*

Er wordt gemotiveerd aangegeven hoe de organisatie zich tegenover het gebruik van informatietechnologie opstelt. Hierbij wordt gebruik gemaakt van de criteria:

1. innovatieve houding op IT-gebied (mate van voorop lopen bij het uitdenken en invoeren van informatietechnologische vernieuwingen);
2. de snelheid waarmee nieuwe producten op het gebied van IT worden aangeschaft;
3. de mate waarin men in staat is om IT voorzieningen zelf te installeren, beheren en programmeren (blijkend uit het al dan net veel zelf doen op dit gebied);
4. de snelheid waarmee verouderde IT-voorzieningen worden vervangen.

Hierbij wordt gebruik gemaakt van de typering:

- innoverende houding;
- afwachtende houding;
- verzetshouding;

Voorbeeld

Bij Organisatie Z is sprake van een afwachtende houding op het gebied van informatietechnologie:

1. men neemt innovaties over als ze passen in het bedrijfsbeleid;
2. nieuwe producten worden alleen aangeschaft als ze niet te kostbaar zijn en niet al te moeilijk te installeren; men blijft wel actief op de hoogte van nieuwe producten;
3. men doet bepaalde dingen wel zelf en heeft in het verleden ook één klein informatiesysteem ontwikkeld; men besteedt weinig uit;
4. producten worden na een normale economische levensduur vervangen

6.4 De verantwoordelijkheidsverdeling op het gebied van informatiesystemen

Er wordt aangegeven hoe de verantwoordelijkheden op het gebied van IT in termen van procesbeheer, informatiebeheer en systeembeheer liggen.

Voorbeeld

Het *procesbeheer* ligt voor het inkoop- en assemblageproces bij de inkoper en voor het verkoopproces bij de verkopers.

Het *informatiebeheer* is bij JEEKAA niet duidelijk toegewezen. Er is geen primair verantwoordelijke voor het onderhoud van de diverse bestanden aangewezen.

Bij JEEKAA is een de verkopers belast met het *systeembeheer*. Dit houdt in dat hij dagelijks aan het einde van de werkdag een back-up maakt en maandelijks bestanden opschoont. Verder zorgt hij voor het installeren van nieuwe programmatuur. Het apparaat-technisch onderhoud wordt uitbesteed.

6.5 Typologie Starreveld

De organisatie kan met behulp van de door Starreveld (1981) gegeven typologie worden gekarakteriseerd (Zie bijlage 1).

Voorbeeld

JEEKAA valt hoofdzakelijk onder de typen:

- een huishouding die voor de markt werkt (type 100);
- een bedrijf met een overwegende doorstroming van eigen goederen (type 110);
- een bedrijf zonder technisch omzettingproces (type 111);
- een handelsbedrijf dat in hoofdzaak aan particulieren (finale consumenten) levert (type 111.2).

Het zeer beperkte assembleren dat bij JEEKAA gebeurt valt onder:

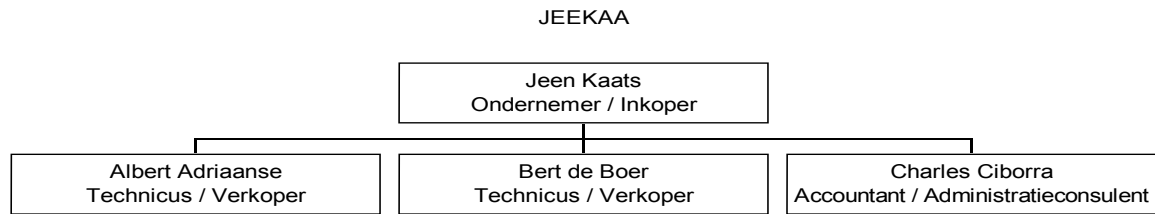
- een huishouding die voor de markt werkt (type 100);
- een bedrijf met een overwegende doorstroming van eigen goederen (type 110);
- een industrieel bedrijf (type 112);
- een bedrijf met seriestuk- en stukproductie (type 112.3);
- een bedrijf met stukproductie (type 112.31).

6.6 Organigram

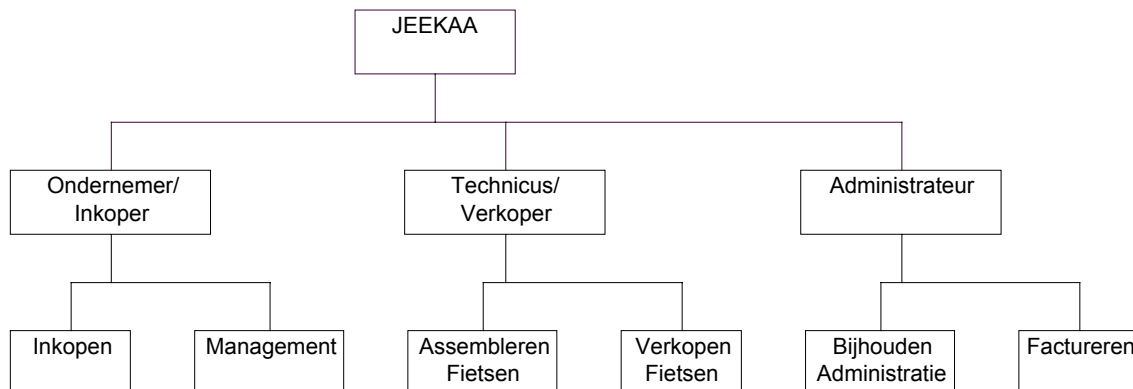
In een *organigram* geven we de functionele structuur van de organisatie weer in drie lagen: de organisatie, de functies en de onder de functies vallende taken.

Voorbeeld

Bij JEEKAA ziet de personele bezetting er als volgt uit.



Op grond van de gegevens over deze personele bezetting kunnen we een organigram maken waarin drie lagen voorkomen: organisatie, functie, en taak. We kunnen dit organigram tekenen met System Architect, module Auto Decomposition. Dit diagram is met kopiëren en plakken (**Edit| Select All; Edit| Copy; Edit| Paste** in Word) vanuit System Architect naar MS Word overgehaald. Het diagram kan in MS Word eventueel verder worden geëdit.



6.7 Bedrijfsprocessen

Het belangrijkste bedrijfsproces kan puntsgewijs worden beschreven. Zie het hieronder gegeven voorbeeld.

Voorbeeld

Bij JEEKAA kunnen de volgende vier bedrijfsprocessen worden onderkend:

- A. algemene verkooproutine:
 - a. bestellen fietsen (trigger: het is maandag) (actor: inkoper):
 - . opnemen voorraad;
 - . opnemen aangevraagde fietsen;
 - . invullen bestelformulier;
 - . bewaren kopie bestelformulier;
 - b. arriveren fietsen (trigger: fietsen komen binnen) (actor: inkoper):
 - . bewaren vrachtbon;
 - . maken aantekening op bestelformulier (soms);
 - . voorzien fietsen van prijs en opstellen in winkel;
 - c. bewaren inkoopfactuur (trigger: binnenkomen factuur) (actor: inkoper);
 - d. verkopen fietsen (trigger: klant wil fiets kopen) (actor: verkoper):
 - . ter hand stellen van fiets;
 - . invullen verkoopfactuur;

- . in ontvangst nemen verkoopbetaling;
 - . doorslag 1 verkoopfactuur bewaren;
 - . doorslag 2 verkoopfactuur naar verzekering;
 - e. betalen inkoop (trigger: het is maandag) (actor: inkoper).
- B. verkopen naar aanleiding van wens klant:
- a. noteren vraag klant om fiets (trigger: klant wil fiets kopen uit catalogus) (actor: verkoper);
 - b. (zie verder) verkopen op voorraad.
- C. annuleren bestellingen:
- a. noteren annulering door klant (trigger: klant annuleert) (actor: verkoper);
 - b. annuleren bestelling bij leverancier (trigger: het is maandag) (actor: inkoper).
- D. beantwoorden vragen om informatie:
- a. noteren vraag om informatie (trigger: klant, politie of verzekering wil framenummer weten) (actor: verkoper);
 - b. beantwoorden vraag om informatie (trigger: het is maandag) (actor: verkoper).

6.8 Gegevensstromen en bestanden

Bij het aspect gegevensstromen en bestanden worden puntsgewijs opgesomd:

- de ingaande en uitgaande berichten en de aanwezige bestanden;
- de aanwezige informatiesystemen.

Zie voorts onderstaand voorbeeld.

Voorbeeld

De ingaande berichten bij JEEKAA zijn (met tussen haakjes de organisatie die de bron is van deze berichten):

- orderbevestiging (leverancier);
- fiets + vrachtbon (leverancier);
- inkoopfactuur (leverancier);
- verkoopbetaling (klant);
- order klant (klant);
- vraag over framenummer (derden).

De uitgaande berichten bij JEEKAA zijn (met tussen haakjes de organisatie die de bestemming is van deze berichten):

- bestelformulier (leverancier);
- fiets (klant);
- verkoopfactuur (klant);
- inkoopbetaling (leverancier);
- antwoord (derden).

De bestanden (ordners, stapels documenten, kaartenbakken, computerbestanden, fysieke verzamelingen die als informatiebron worden gebruikt, e.d. die nu worden bijgehouden en/ of gebruikt bij JEEKAA zijn:

- kopie bestelformulieren;

- fietsen;
- vrachtbonnen;
- inkoopfacturen;
- fietsen met prijsgegevens;
- doorslagen van verkoopfacturen;
- notities dat fiets is verkocht;
- verkoop-betalingsafschriften;
- inkoop-betalingsafschriften;
- notities met orders klanten (o.a. met NAW gegevens klant);
- notities met vragen van derden.

6.9 De functionaliteit van het informatiesysteem

De functies van het informatiesysteem worden opgesomd geordend volgens de volgende indeling:

- ondersteuning gebruikerstaken;
- ondersteuning informatiebeheer;
- ondersteuning systeembeheer.

Voorbeeld

Bij het gebruik van het fietsinformatiesysteem is sprake van sterk interactieve taakondersteuning (een soort kantoorautomatisering). Verder levert het informatiesysteem zelfstandig adviezen over de te bestellen fietsen.

De systeemfuncties van fietsinformatiesysteem XX zijn:

Ondersteuning gebruikerstaken:

1. spoedbestelling:
 - opvragen/invullen klantgegevens;
 - kredietwaardigheid nagaan;
 - invullen bestelformulier;
 - verzenden bestelformulier naar leverancier;
2. annuleren bestelling:
 - opvragen klantgegevens;
 - opvragen bestelgegevens;
 - invullen annulering;
 - verzenden annulering naar leverancier;
3. beantwoorden vragen:
 - opvragen gegevens fietsen;
4. bestellen fietsen:
 - opvragen gegevens bestelkosten, voorraadkosten en maken vraagprognose;
 - toepassen bestelformule;
 - invullen bestelformulier;
 - verzenden bestelformulier naar leverancier;
5. arriveren fietsen:
 - opvragen bestelgegevens;
 - invoeren fietsgegevens;
6. verkopen fietsen:
 - opvragen/invullen klantgegevens;

- kredietwaardigheid nagaan;
 - opvragen fietsgegevens;
 - invullen verkoopformulier;
 - verzenden verkoopformulier naar financiële administratie;
7. betalen inkoop:
- opvragen wekelijkse bestelgegevens, fietsgegevens en verkoopgegevens;
 - verzenden wekelijkse gegevens naar financiële administratie.

Informatiebeheer:

8. klantgegevens beheren
9. algemene fietsgegevens beheren
10. leveranciersgegevens beheren
11. bijwerken algemene gegevens
12. inkoopformule en verwante regels beheren

Systeembeheer:

13. maken back-up
14. opschonen bestanden
15. beveiliging

6.10 Het gebruikersinterface van het informatiesysteem

Het gebruikersinterface van het te beschrijven informatiesysteem wordt beoordeeld op de volgende punten:

- consistentie;
- presentatie en huisstijl;
- duidelijkheid van de werkomgeving;
- in hoeverre de gebruiker aan het stuur zit;
- kwaliteit van de feed-back;
- foutafhandeling;
- of rekening wordt gehouden met de beperkingen van het geheugen van de gebruiker.

Voorbeeld

Het informatiesysteem X scoort als volgt op de beoordelingspunten van het gebruikersinterface:

<i>Nr</i>	<i>Beoordelingspunt</i>	<i>Waardering</i>	<i>Reden</i>
1.	consistentie	goed	conform Windows conventies
2.	presentatie; huisstijl	matig	een rommelig geheel en niet consistent, saai kleurgebruik, teveel lettertypes door elkaar; sommige letters zijn slecht leesbaar op een kleine monitor
3.	duidelijkheid werkomgeving	slecht	hoe een opdracht moet worden ingevoerd is uitermate onduidelijk (in de browse-modus naar het einde van de lijst gaan en daar allerlei duistere handelingen uitvoeren)
4.	initiatief bij de gebruiker	matig	geen undo mogelijkheid
5.	feed-back	redelijk	maar er is soms dead air
6.	foutafhandeling	slecht	er wordt geen enkele ondersteuning gegeven
7.	beperkingen van het geheugen	matig	veel ingewikkelde functietoetsen en ingewikkelde menubomen

6.11 De opbouw van het informatiesysteem

De opbouw van het informatiesysteem uit kleinere subsystemen of modules wordt weergegeven door een geordende opsomming van deze modules. Daarbij wordt de volgende indeling gebruikt:

- documenten (op scherm en op andere media);
- gegevensbank;
- taakgerichte modules;
- besturingselementen.

Voorbeeld

Het informatiesysteem S is uit de volgende modules opgebouwd:

Documenten:

- editor (scherm en op papier)
- rapportage (van diverse soorten; op scherm en op papier)
- toolbox
- selectieschermen ter ondersteuning menu (van diverse soorten; alleen op scherm)
- schermen voor de definitie van nieuwe symbolen

Gegevensbank:

- relationele gegevensbank (dBASE)
- bestanden met tekeningen

Taakgerichte modules:

- Auto-decomposition
- Character Screen
- Data Flow Gane & Sarson
- Data Flow Ward & Mellor
- Data Flow Yourdon & De Marco
- Decomposition
- Entity Relation
- Flow Chart
- IDEF0
- IDEF1X
- Logical View
- Menu
- OMT Data Flow
- OMT Event Flow
- OMT Event Trace
- OMT Object Model
- OMT State
- SQL Server Physical
- SSADM Context
- SSADM Data Flow
- SSADM Dialogue Structure
- SSADM Document Flow
- SSADM Effect Corr.
- SSADM Entity Life History
- SSADM I/O Structure
- SSADM Logical Database Press
- SSADM Menu Structure
- SSADM Physical Database Prcss
- SSADM Resource Flow
- SSADM Update Process
- State Transition
- State Transition Ward & Mellor
- Structure Chart

Besturingselementen:

- menustructuur
- popup-menu rechter muisknop
- buttons op toolbox
- check boxes in selectieschermen ter ondersteuning menu
- invoervelden en check boxes in schermen voor invoer nieuwe symbolen

7. Actor-interactiemodellen (B1-modellen)

7.1 Actor-interactiemodellen (B1-modellen)

Bij actor-interactiemodellen (B1-modellen, waarbij de B staat voor bedrijf) kijken we naar de systemen en actoren die van belang zijn in het te bestuderen probleemgebied en hun interacties. De afbakening van interacties berust op een bedrijfskundig en systeemtheoretisch begrip van transactie en transformatie. De afbakening van interacties is ook gebaseerd op het standaardpatroon van een interactie volgens de DEMO methode.

De beschrijving van interacties blijft functioneel, het gaat over wat een organisatie doet zonder te kijken naar hoe dat gebeurt. Systemen, actoren en interacties worden in hun onderlinge relatie, maar zonder besturing en tijd afgebeeld.

Het blijkt dat interacties, mits goed in kaart gebracht, een goede basis bieden voor de indeling van de functies die het informatiesysteem moet kunnen vervullen.. De actor-interactiemodellen die wij tekenen, de actor-interactiediagrammen, maken gebruik van de OMT Data Flow notatie.

Op het gebied van actoren en interacties kunnen *actor-interactiediagrammen* op twee reticulatieniveaus worden getekend: op het niveau van organisaties die met elkaar interacteren (diagram 'Organisaties en Transacties', besproken in Paragraaf 7.2.) en op het niveau van actoren binnen een organisatie die met elkaar en met externe systemen communiceren (diagram: 'Actoren, Transacties en Transformaties', besproken in Paragraaf 7.3.).

7.2 Organisaties en transacties

Het eerste diagram dat we moeten maken heet Organisaties en Transacties (afkorting ORGAN). Hierin worden de organisaties in het probleemgebied weergegeven en de transacties tussen deze organisaties in termen van hun samenstellende interacties. In het diagram worden vier soorten interacties onderscheiden: levering van producten (P), levering van diensten (services) (S), creëren of overdragen van rechten (R) en betaling van producten of diensten (financiën) (F).

Het diagram dat we voor de fietsenzaak JEEKAA gaan maken heet JKORGAN1. Het wordt met System Architect, module OMT Data Flow, gemaakt. Bij JEEKAA zijn de volgende organisaties te onderscheiden:

- JEEKAA zelf;
- Leverancier (een abstractie van alle leveranciers die JEEKAA heeft, waaronder fietsenfabrieken);
- Klant (een abstractie van alle klanten die JEEKAA heeft);
- Derde (een abstractie van alle Derden die JEEKAA diensten verleent, met name het geven van informatie over framenummers aan politie en verzekeringsmaatschappijen)

Deze organisaties worden met het Actor symbool (een vierkant) getekend.

De volgende transacties zijn te herkennen:

- Inkoop (JEEKAA-Leverancier);

- Verkoop (JEEKAA-Klant):
- Beantwoording Vragen (JEEKAA-Derde).

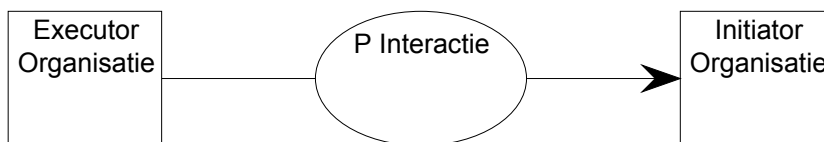
Deze transacties vallen weer uiteen in de volgende interacties:

- Inkoop Levering (P);
- Inkoop Betaling(F);
- Verkoop Levering (P);
- Verkoop Betaling (F);
- Beantwoording vragen (S).

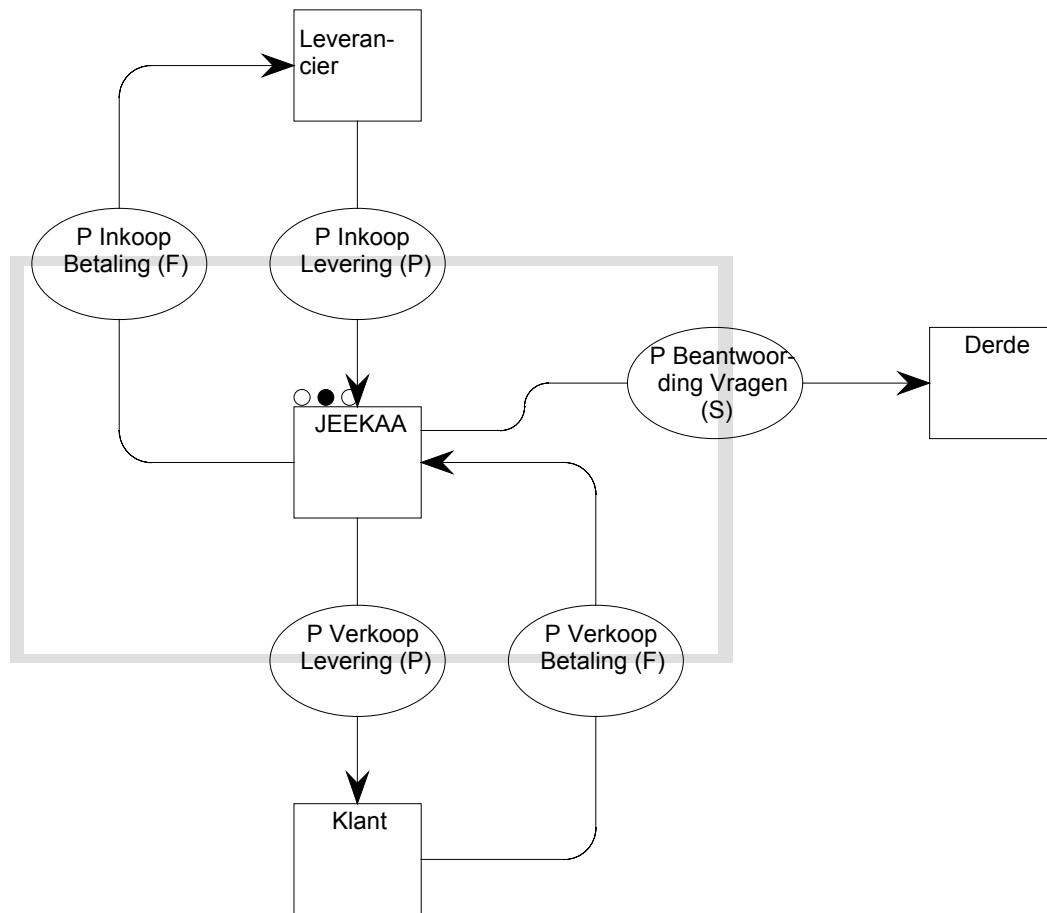
De beantwoording van vragen heeft geen tegenprestatie.

Merk op dat de namen van de interacties steeds bestaan uit de naam van de transactie en zo nodig een verdere specificatie van de interactie. Voor de verdere specificatie van de interactie wordt bij voorkeur de 'ing' uitgang gebruikt. Er wordt bij de naamgeving geredeneerd vanuit het systeem dat de processen uitvoert (en zo de bron van de bijbehorende productstroom, dienststroom of geldstroom is). Daarom spreken we van 'Inkoop Levering' en niet van 'Inkoop Bestelling'.

Een interactie wordt getekend als combinatie van een lijnstuk, een ellips (**Process** symbool) en een pijl. Het lijnstuk zit tussen de executor-organisatie en de ellips; de pijl wijst van de ellips naar de initiator-organisatie. (Pijlpunten kunnen aan en uit gezet worden met de **Associative Properties** van de **Data Flow** symbolen die we gebruiken om de lijnstukken en pijlen te tekenen). Een beetje schuiven met de symbolen is vervolgens nodig om een net geheel te krijgen.



Het resultaat voor JKORGAN1 ziet er als volgt uit. De systeemgrens is in MS-Word bijgetekend.



7.3 Actoren, transacties en transformaties

Per organisatie wordt nu de interne structuur beschreven door middel van actoren en de interacties waarvoor zij verantwoordelijk zijn. Behalve de al eerder onderscheiden transactieprocessen komen er nu de interne bedrijfsprocessen, de transformatieprocessen, bij. Het te tekenen diagram heet 'Actoren, Transacties en Transformaties' (Afkort ACTOR). Het tekenen gebeurt weer met System Architect, module OMT Data Flow.

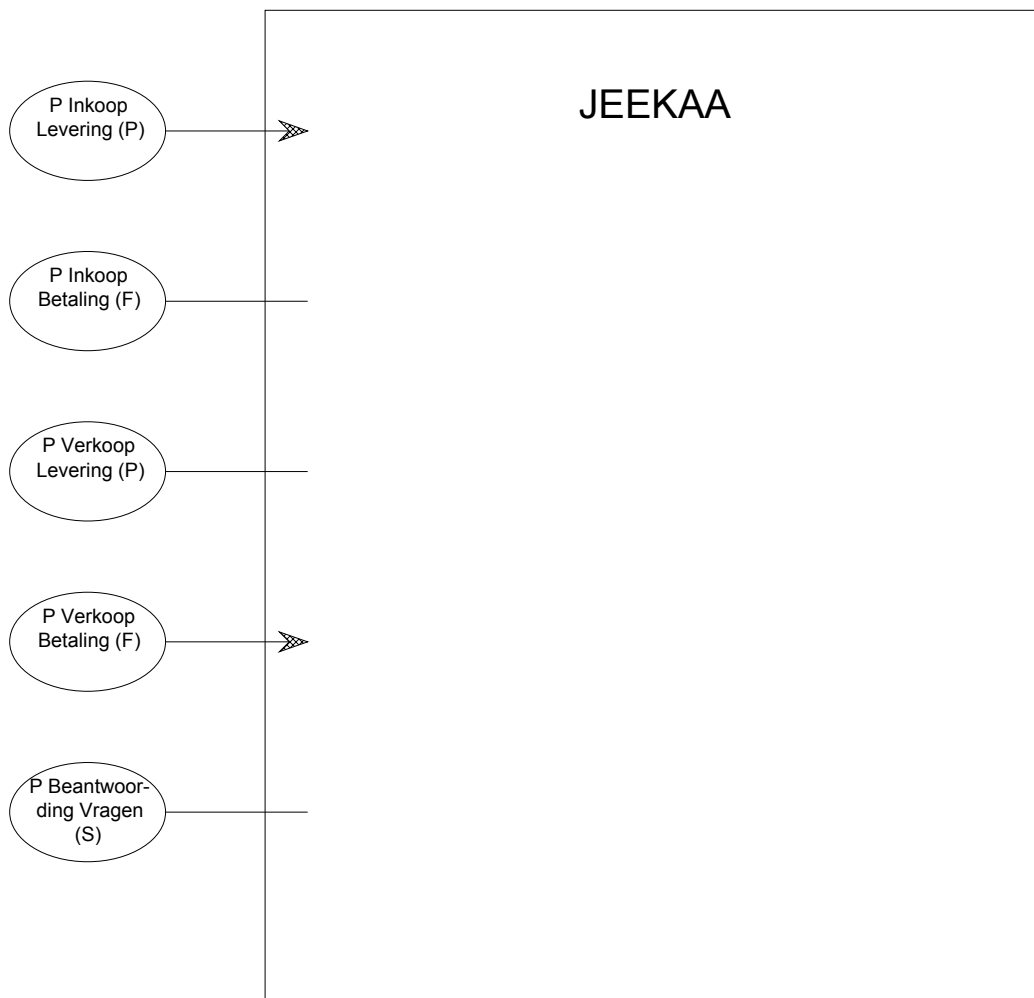
Een actor is een deelsysteem dat verantwoordelijk is (1) voor de besturing, of (2) voor de besturing en uitvoering (executie), van een interactie. Als we uitgaan van de tot dusverre onderscheiden interacties, zijn binnen JEEKAA de volgende actoren te onderscheiden:

- inkoper;
- betaler inkoop;
- verkoper;
- factureerder verkoop;
- beantwoorder vragen.

Vervolgens zijn er de transformatieprocessen waarbij producten of diensten worden gemaakt. De bij deze transformatieprocessen behorende interacties hebben de afkorting T van transformatie. Bij JEEKAA is dit de transformatie 'Assembleren Fietsen', de bijbehorende interactie heet 'Assembleren Fietsen (T)', en de bijbehorende actor heet:

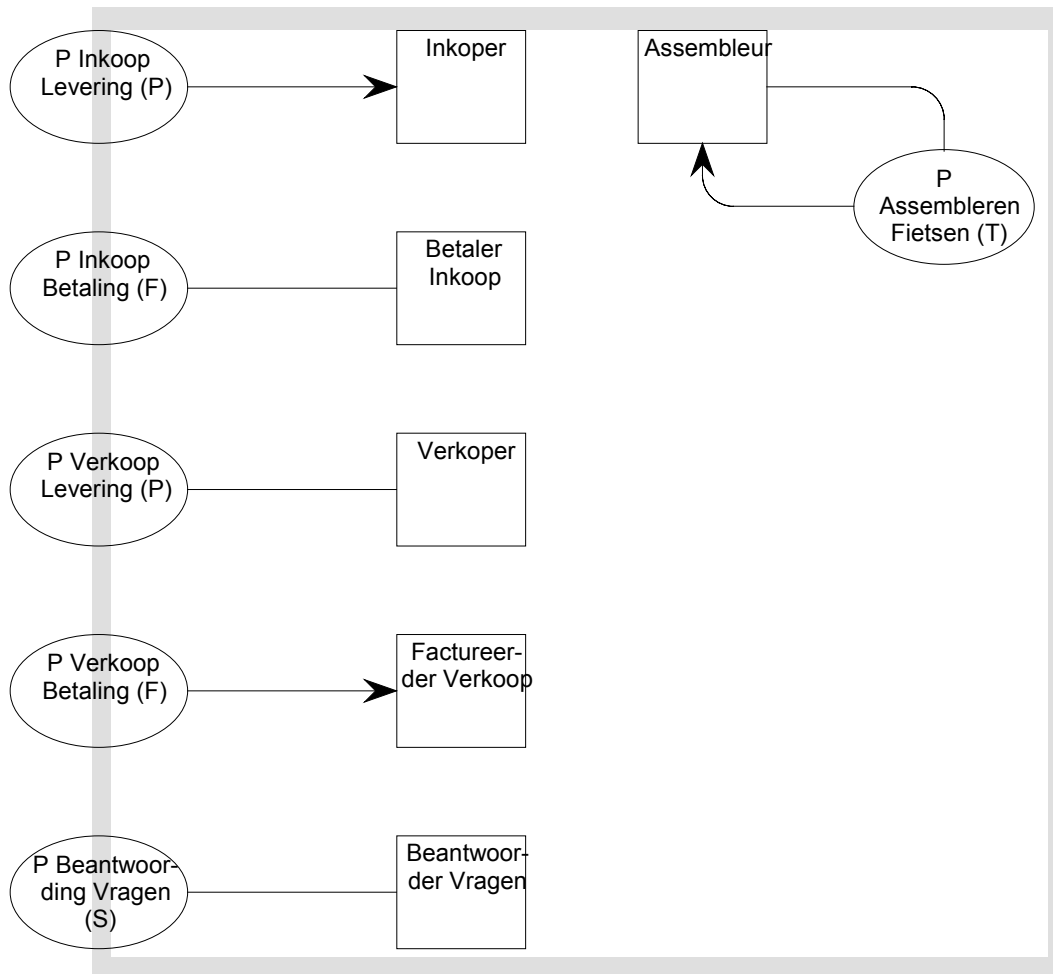
- assembleur.

Voor het tekenen van dit JKACTOR1 diagram maken we gebruik van de *uitvergrotingsmogelijkheid* van System Architect. Het is mogelijk een organisatie (actor) te nemen en daarop in te zoomen (te reticuleren). Voor de interne structuur van de gekozen actor kunnen we vervolgens een nieuw diagram tekenen. Open het ORGAN diagram (JKORGAN1) en klik met de <rechter muisknop> op de centrale organisatie (JEEKAA) en er verschijnt een pop-up menu waaruit **Child Create** moet worden gekozen. Vul de goede naam voor het ACTOR diagram in en klik enkele keren op accoord. Er verschijnt dan een diagram waarin de interne structuur van de centrale organisatie (JEEKAA) kan worden getekend. Voor JEEKAA ziet dit er als volgt uit:



Hierna worden de verschillende actoren getekend, en het een en ander wordt netjes gerangschikt. Om het een en ander beter te zien wordt aangeraden terug te schakelen naar 1 op 1 vergroting door middel van de menukeuze **View | Actual Size**. Het is ook aan te raden de door System Architect getekende systeemgrens weg te halen, of de optie **Set| Preferences| Immediate Auto Routing** uit te schakelen, omdat de tekening zich anders heel narrig gaat gedragen. De systeemgrens kan later in MS Word worden bijgetekend.

Het resulterende diagram ziet er als volgt uit:



8. Bedrijfsprocesmodellen (B2-modellen)

8.1 Bedrijfsprocesmodellen (B2-modellen)

Bedrijfsprocesmodellen (B2- modellen waarbij de B2 staat voor de B van bedrijf en de B van besturing) zijn modellen van de besturing van transacties en transformaties. Bij bedrijfsprocesmodellen modellen speelt besturing, tijd en volgorde een rol. Doordat de notie van volgorde is geïntroduceerd kunnen we spreken van bedrijfsprocessen die worden gemodelleerd.

Een geheel van samenhangende interacties kan worden beschouwd als *bedrijfsproces*: een samenhangend geheel van *toestanden* (waarin *activiteiten* kunnen worden gedaan) en *impulsen* die leiden tot toestandsovergangen (die corresponderen met *acties*). Een bedrijfsproces omvat niet alleen de samenhangende interacties, maar ook de 'lijm' die voor deze samenhang moet zorgen: de impulsen die van de ene interactiefase naar de andere leiden.

Tot dusverre hebben we diagrammen getekend waarin de besturing niet expliciet was afgebeeld. Meestal was tijd en volgorde ook niet afgebeeld omdat volgorde het gevolg is van het lopen van een besturingsimpuls van het ene proces naar het daaropvolgende proces. Bij de beschouwing van bedrijfsprocessen kijken we wel expliciet naar besturing en volgorde. Procesdiagrammen zijn altijd toestandsdiagrammen waarin toestanden en toestandsovergangen worden weergegeven. We tekenen ze met de System Architect modules OMT State en OMT Event Trace. In de module OMT Event Trace is de toestand van het systeem een samenstel van de toestanden van de deelnemende actoren die in de vorm van tijdslijnen worden weergegeven..

Bedrijfsprocessen bestaan uit samenhangende interacties. Allereerst wordt gekeken naar de samenhang van interacties op een grofmazig niveau. Interacties hebben verbanden waarbij de volgorde belangrijk is: het aflopen van de ene interactiefase is voor een andere interactiefase een impuls om te beginnen (zie Paragraaf 8.2.).

Het bedrijfsproces wordt in meer detail geanalyseerd op grond van tijdslijnen van actoren en de impulsen die de actoren elkaar zenden Dit bedrijfsprocesdiagram kan worden gebruikt om punten op te sporen waar het bedrijfsproces (soms onnodig) moet wachten op de voortgang van de activiteiten van één der betrokken actoren of systemen (zie Paragraaf 8.3.).

De interne structuur van een interactie kan worden beschreven op het niveau van activiteiten en acties, waarbij ook uitzonderingen op de standaard gang van zaken aan de orde komen(zie Paragraaf 8.4.).

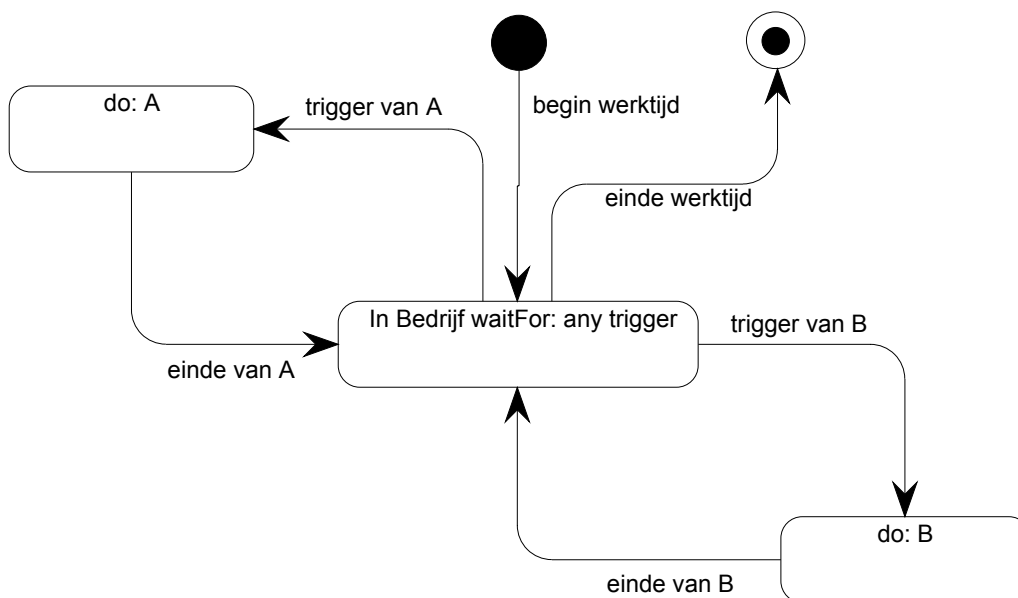
Mede op grond van het bedrijfsprocesdiagram kan een beoordeling van het bedrijfsproces plaatsvinden (zie Paragraaf 8.5.).

8.2 Verbanden tussen interacties

De verbanden tussen de interacties in een systeem kunnen door middel van een toestandsdiagram worden beschreven. In een toestanddiagram worden toestanden weergegeven door een afgeronde rechthoek en toestandsovergangen (impulsen) door pijlen.

De naam van dit diagram is INTER. Een bepaalde interactie of interactiefase is op te vatten als een activiteit. Hier wordt even in herinnering gebracht dat activiteiten, in tegenstelling tot acties, een duur hebben. Een actor is even, of langer durend, bezig met een activiteit en dit correspondeert met een toestand 'bezig met activiteit X'. Dit wordt in de naam van de toestand aangegeven door middel van **do: activiteit X**. Een activiteit geeft na afloop een impuls die een andere activiteit laat beginnen. Soms is het van de waarde van bepaalde parameters afhankelijk of een vervolgactiviteit zal starten of niet. De condities waaronder een vervolgactiviteit start worden weergegeven tussen vierkante haken []. Het START-symbool is een zwarte stip. Het STOP-symbool is een zwarte stip met een cirkel er om heen (een z.g. Bull's Eye, bekend van het dart spel). Deze diagrammen worden met System Architect, module OMT State, getekend.

Het algemene patroon van een toestandsdiagram voor een organisatie ziet er als volgt uit. We beginnen bij het START-symbool (zwarte stip). Na de gebeurtenis 'begin werktijd' komt de organisatie in de toestand 'In Bedrijf'. Nadat de gebeurtenis 'einde werktijd' heeft plaatsgevonden komen we bij het STOP-symbool. Tussen begin werktijd en einde werktijd kunnen gebeurtenissen zorgen voor het in gang zetten van interacties. Als de organisatie bezig is met Interactie A, is ze in de toestand **do: A**. Als er in een toestand moet worden gewacht wordt dit weergegeven door **waitFor: X**, waarbij X het soort impuls is waarop wordt gewacht. De acties (impulsen) worden weergegeven door pijlen; naast de pijl staat de naam van de actie. In onderstaande figuur zijn twee activiteiten (A en B) afgebeeld die in gang kunnen worden gezet. Er zijn er natuurlijk meer mogelijk.



Het diagram 'Verbanden tussen interacties' (INTER) ziet er voor JEEKAA als volgt uit. Het diagram heet JKINTER1. Let op hoe de interactiefasen zijn weergegeven: na de naam van de interactie een schuine streep met de letter van de interactiefase. Het opnemen van interactiefasen in dit diagram moet alleen worden gedaan als het echt nodig is.

Impulsen die met de tijd te maken hebben (in het algemeen gaat het hier om activiteiten die worden getriggerd op een 'vast' periodiek tijdstip) krijgen een naam die met een T. begint. Een voorbeeld is de gebeurtenis *T. elke maandag*.

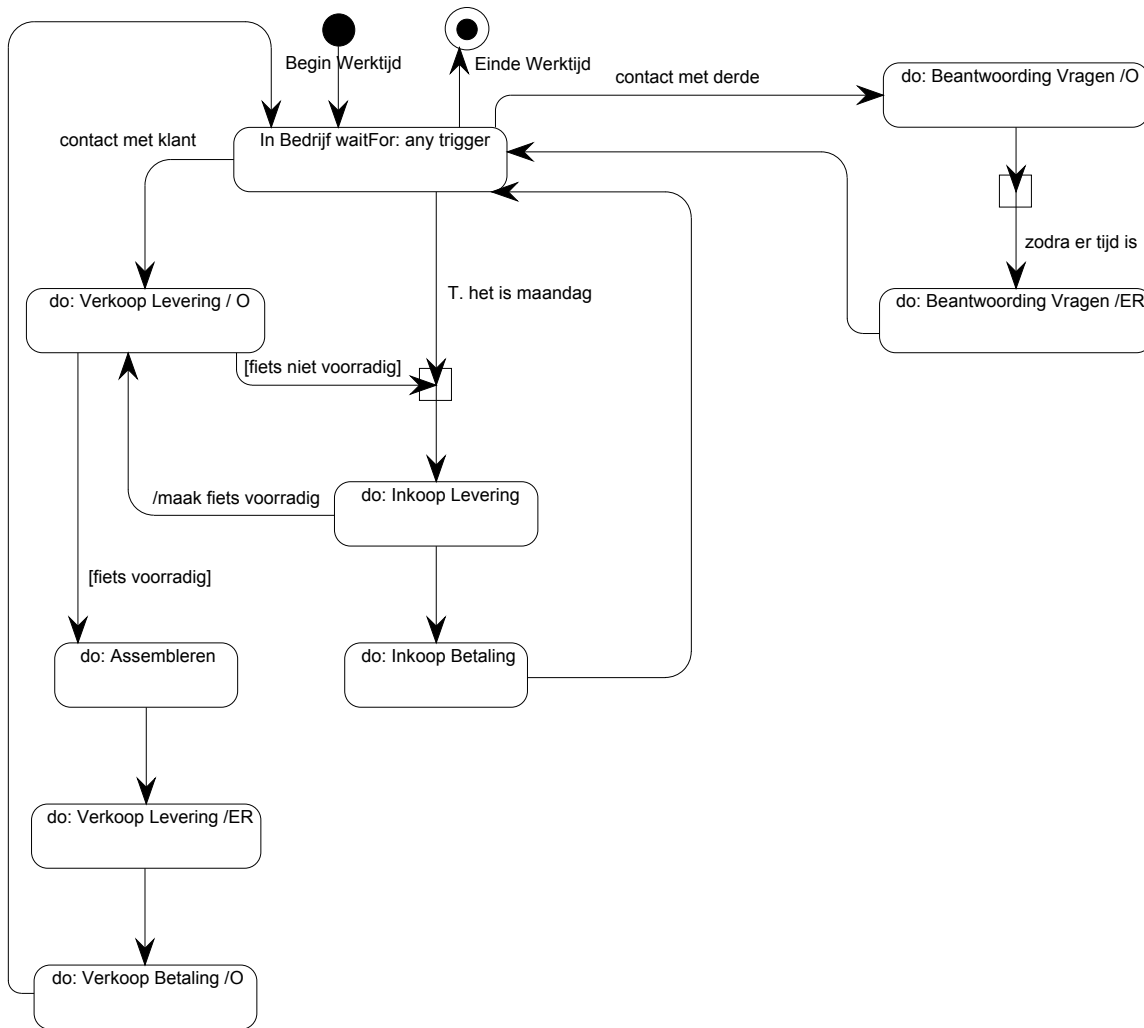
In het geval van JEEKAA moet een fiets worden besteld als die niet voorradig is. Deze *conditie [fiets niet voorradig]* wordt tussen vierkante haken weergegeven. Met een conditie **[A]** correspondeert meestal een conditie **[niet A]** in het diagram. In dit geval is er ook een conditie **[fiets voorradig]**.

Soms moet aan meerdere condities worden voldaan voordat een bepaalde activiteit kan beginnen. Er moeten dan meerdere acties hebben plaatsgevonden. Bijvoorbeeld in het geval dat een fiets niet voorradig is kan vanuit de Verkoop Levering /E (executiefase) niet direct worden overgegaan tot de Inkoop Levering /O (opdrachtfase). Er moet eerst worden gewacht worden tot het weer maandag is, want dan vinden de bestellingen altijd plaats. De combinatie van beide conditionele acties wordt weergegeven door een **AND** blokje. Vanuit het **AND** blokje wordt vervolgens de Inkoop Levering getriggerd. Bij het tekenen van de verschillende pijlen maakt System Architect het AND blokje soms onzichtbaar. Het is het beste om er dan maar een nieuw blokje overheen te tekenen.

De pijlen die acties (gebeurtenissen, toestandsovergangen) weergeven kunnen zich eventueel splitsen. Dit is van belang als een actie als conditie voor meer dan één activiteit moet dienen. De oorspronkelijke pijl loopt dan naar een **AND** blokje. Vanuit het **AND** blokje lopen er dan twee of meer pijlen naar de activiteiten die moeten worden gestart.

Soms kan er als een bepaalde activiteit is afgelopen niet meteen begonnen worden aan de vervolgactiviteit. Een voorbeeld is dat aan de uitvoering van de beantwoording van vragen (*Beantwoording Vragen /E*) niet meteen na afloop van het noteren van de vraag (*Beantwoording Vragen /O*) wordt begonnen, maar pas zodra er tijd is. We tekenen dan een AND blokje na de actie die voortkomt uit het aflopen van de activiteit. Vanuit het AND blokje tekenen we een tweede actiepijl naar de vervolgactiviteit met erbij de gebeurtenis die zorgt voor het starten van de vervolgactiviteit.

De AND blokjes geven een toestand weer waarin het systeem op een bepaalde gebeurtenis moet wachten. Deze wachttoestanden corresponderen over het algemeen met voorraden berichten of producten. Merkt men dergelijke voorraden op dan is dat vaak een aanwijzing voor een dergelijke AND tussentoestand.



8.3 Tijdslijnen van actoren en acties

Bedrijfsprocessen kunnen worden geanalyseerd met een diagram 'Tijdslijnen van actoren en acties' (bedrijfsprocesdiagram), waarin de tijdslijnen van actoren en hun acties worden afgebeeld (naam: BPROC, te tekenen met System Architect module OMT Event Trace). Een bedrijfsprocesdiagram is vooral van belang bij het analyseren waar de activiteiten van actoren onnodig moeten wachten. In een bedrijfsprocesdiagram wordt de toestand van het geheel niet door een afgeronde rechthoek weergegeven, maar als een combinatie van de toestanden van de deelnemende actoren op een bepaald tijdstip. Dit is als het ware een horizontale dwarsdoorsnede door alle tijdslijnen op een bepaald ogenblik; de verzameling snijpunten van dwarsdoorsnede en tijdslijnen correspondeert dan met de toestand van het totale systeem.

In het bedrijfsprocesdiagram zijn er van boven naar beneden lopende *tijdslijnen* voor elk der actoren, aangegeven door een gestippelde lijn (in OMT Event Trace heten ze **Object**). De tijdslijnen voor de nu bekende actoren kunnen gemakkelijk worden aangemaakt door uit de actor-interactie diagrammen alle actoren te kopiëren naar het nieuwe BPROC diagram, en vervolgens elk van de symbolen met <rechtermuisklik> | **Transform Symbol** naar **Object** over te zetten. De tijdslijn van een actor kan zich splitsen; hij/ zij doet dan parallelle activiteiten.

Impulsen worden weergegeven door (meestal horizontale) pijlen die van de impuls genererende actor naar de impuls ontvangende actor lopen (in OMT Event Trace heten ze **Event**). Impulsen kunnen onder een bepaalde *conditie* plaatsvinden. Deze conditie moet dan tussen vierkante haken bij de pijl worden weergegeven. Een voorbeeld is [**fiets niet voorradig**]. Impulsen kunnen ook gepaard gaan met een bepaalde actie. Deze actie wordt dan achter een schuine streep bij de pijl vermeld. Een voorbeeld is /**maak fiets voorradig**.

De impulsen voor de JEEKAA case kunnen min of meer automatisch uit de standaard structuur van een interactie volgens DEMO (zie Paragraaf 4.3.) worden afgeleid. De standaard namen voor de impulsen in de interactie tussen Inkoper en Leverancier zijn bijvoorbeeld:

- inkoop levering verzoek;
- inkoop levering belofte;
- inkoop levering uitvoering;
- inkoop levering verklaring;
- inkoop levering aanvaarding.

In de meeste gevallen wordt een interactie niet doorbroken doordat eerst een andere interactie moet plaatsvinden. Dit is echter bij de Verkoop Levering niet het geval: in het geval dat een fiets niet voorradig is moet eerst de Inkoop Levering plaatsvinden, en in alle gevallen moet voor de executiefase van de Verkoop Levering eerst het Assembleren plaatsvinden. Deze verbanden tussen interacties blijken uit het INTER diagram (zie Paragraaf 6.4.).

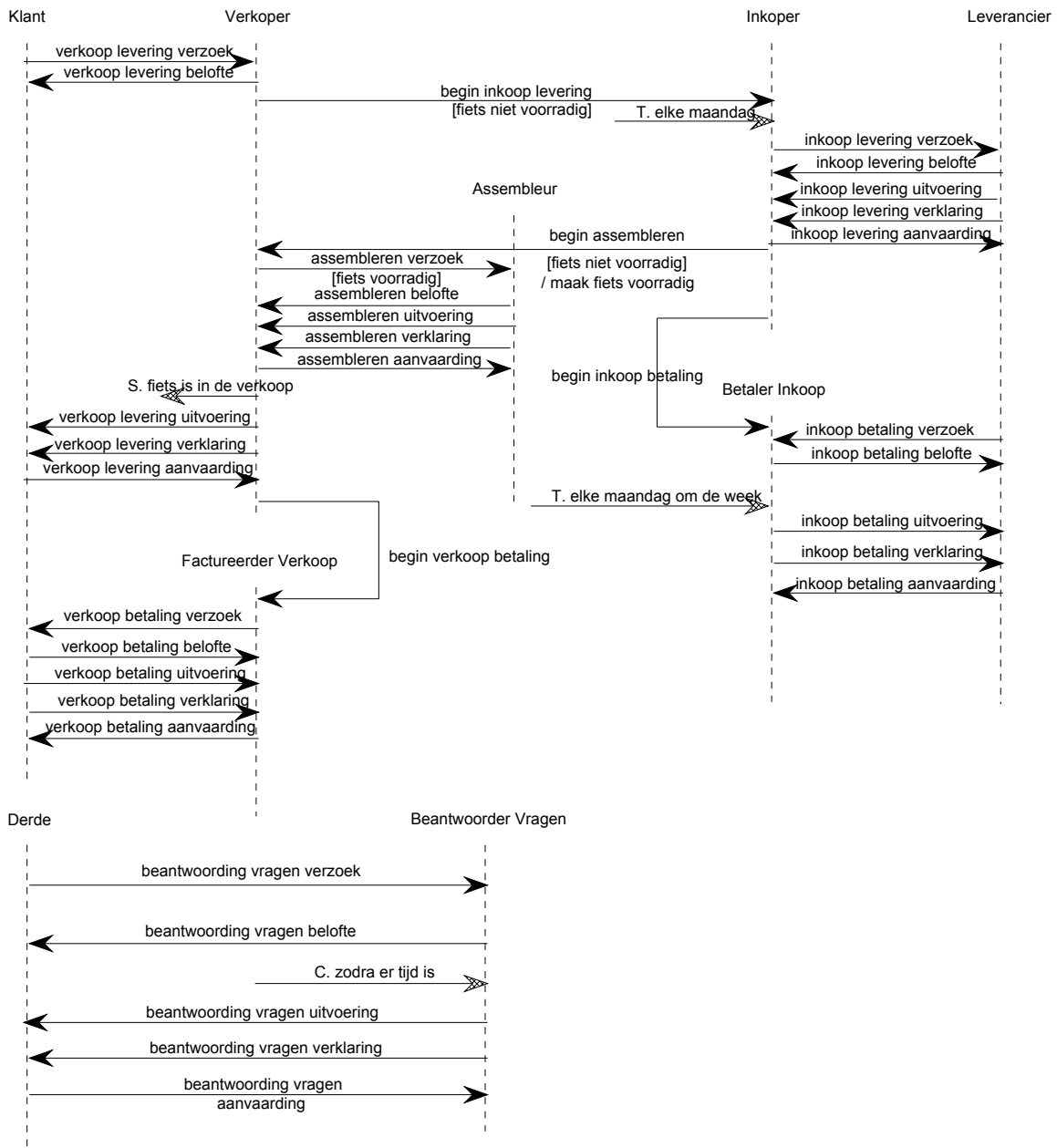
Een *toestand* van een actor kan worden aangeduid door een pijl die vanaf de tijdslijn van een actor wegloopt en niet aan een andere actor vastzit. De naam van de toestand begint met een S. Een voorbeeld is **S. fiets is in de verkoop**.

Twee bijzondere impulsen, die niet door actoren worden gegenereerd zijn de conditie-impuls en de tijdsimpuls. De pijlen van beide impulsen zitten dus maar aan één kant, die van de pijlpunt, aan een actor-tijdslijn vast. De conditie-impuls vindt plaats zodra aan een bepaalde conditie is voldaan. De naam ervan begint met een C. Een voorbeeld is **C. zodra er tijd is**. De tijdsimpuls vindt plaats op een vast tijdstip of met een vaste frequentie,. De naam ervan begint met een T. Een voorbeeld is **T. elke maandag**.

Een actor is in principe steeds bezig met activiteiten op zijn tijdslijn behalve op de punten waar een impuls-pijlpunt de tijdslijn raakt. Op die plaatsen moet de actor wachten tot de impuls heeft plaatsgevonden. Dit betekent dat elk lijnstuk op de tijdslijn tussen twee impulsen correspondeert met een toestand waarin de actor een activiteit wordt uitvoert (**do: activiteit A**). Elk raakpunt aan een pijlpunt op de tijdslijn correspondeert met een toestand waarin de actor moet wachten op een impuls (**waitFor: impuls B**).

Een analyse van een bedrijfsprocesdiagram houdt onder meer in dat men heel kritisch kijkt naar alle punten waar een actor moet wachten. We zien in het diagram voor JEEKAA dat er misschien onnodig moet worden gewacht bij het begonnen met de Inkoop Levering, de Inkoop Betaling Uitvoering, en de Beantwoording Vragen Uitvoering.

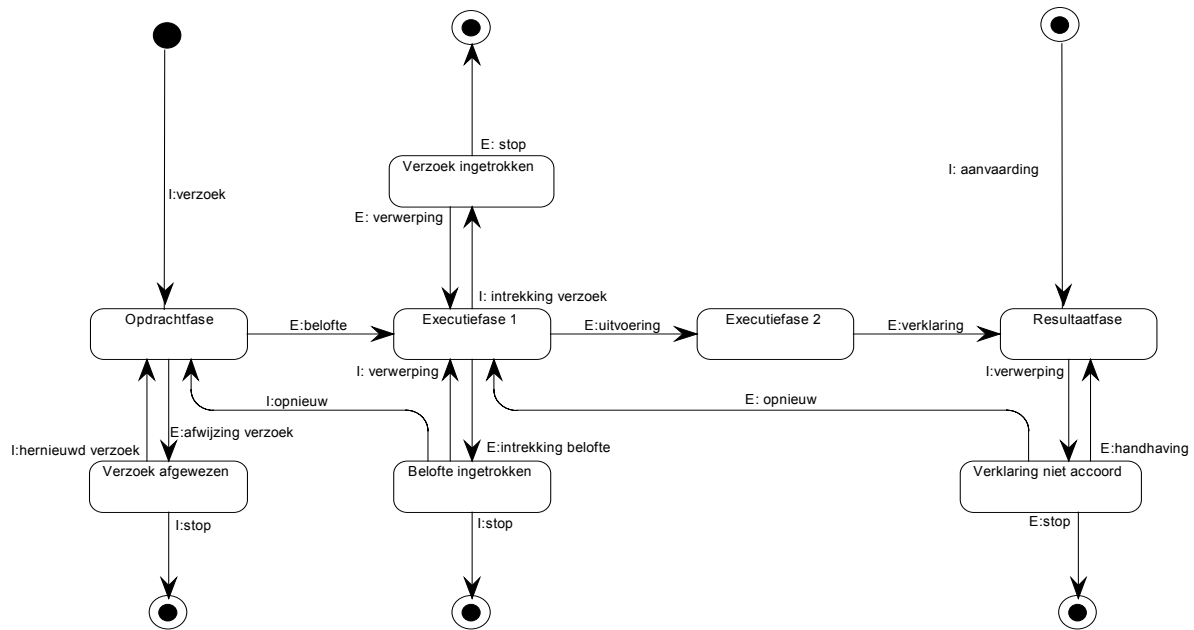
Een bedrijfsprocesdiagram kan worden getekend met de System Architect module OMT Event Trace. Het diagram voor JEEKAA, JKBPROC1, is hieronder weergegeven.



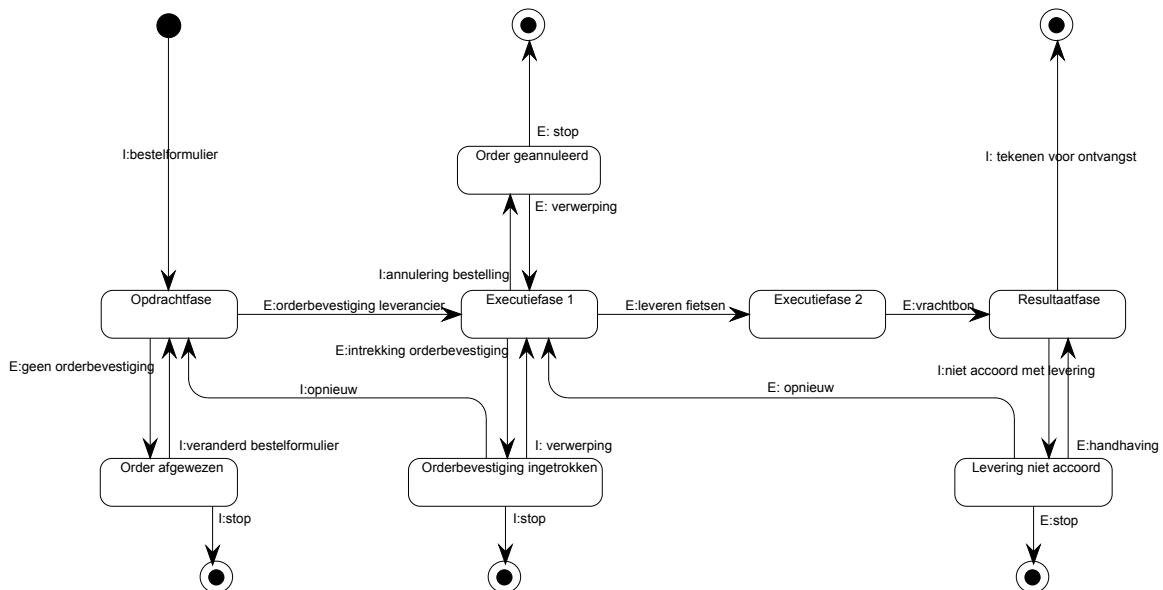
8.4 De interne structuur van een interactie

Per interactie waarvoor dit relevant is, kan een meer gedetailleerd toestandsdiagram worden getekend waarin ook de uitzonderingsacties zijn weergegeven. Dit diagram heet STATE. Het wordt weer met System Architect, module OMT State, getekend.

Het algemene beeld voor een toestandsdiagram voor een interactie volgens de DEMO theorie, vertaald in de OMT State modellering, ziet er als volgt uit:



Toegepast op de JEEKAA interactie Inkoop Levering ziet het toestandsdiagram er als volgt uit:



Merk op dat bij elke actie ook een bericht hoort. Deze berichten spelen een rol in het volgende diagram (zie Paragraaf 9.3).

8.5 Beoordeling van bedrijfsprocessen

In de Business Process Redesign school (Hammer, 1990) wordt bij de beoordeling van bedrijfsprocessen onder meer naar het volgende gekeken:

1. onderzoek de breuklijnen in de bedrijfsprocessen (zoals opnieuw invoeren van dezelfde gegevens) en ruim deze uit de weg;

2. schrap alle materiële voorraden behalve de directe werkvoorraden, de klantorder-ontkoppelpunten en de verzamelpunten;
3. schrap voorraden documenten die met de computer niet meer nodig zijn;
4. probeer complexe processen met veel uitzonderingen te vereenvoudigen door ze te splitsen in een netwerk van eenvoudige, gestandaardiseerde, processen;
5. schrap controle en beheersingsactiviteiten die geen toegevoegde waarde hebben;
6. schrap onnodige stappen in de bedrijfsprocessen (o.a. door verruiming van verantwoordelijkheden kan de overdracht van een taak van actor naar actor veelal vervallen);
7. maak de capaciteit van de organisatie groter door flexibele inzet van personeel (o.a. mogelijk gemaakt door taakverbreding) en parallel werken;
8. zorg voor betere specificaties en snellere feedback zodat werk niet over gedaan hoeft te worden, o.a. door het werk dicht bij de klant te brengen.

Het toepassen van deze principes betekent veelal dat allerlei overbodig werk kan worden geschrapt en dat wachttijden vervallen.

Bij de beoordeling hoe activiteiten worden uitgevoerd is het vaak nodig om de manier van administreren of beslissen zoals het volgens het boekje (bijvoorbeeld Starreveld) hoort te combineren met de nodige pragmatische overwegingen om de nieuwe manier van werken ook praktisch aanvaardbaar/ uitvoerbaar te maken.

In het geval van JEEKAA is het wenselijk om het reageren op een order van een klant te versnellen door een *spoedbestelprocedure* toe te passen. In deze procedure zal een (lichte) toets kunnen worden verwerkt in verband met de kredietwaardigheid van een klant, om bijvoorbeeld bekende wanbetalers uit te sluiten. Ook annuleringen zouden op een dergelijke manier moeten worden verwerkt. Dit betekent dat bij JEEKAA op grond van principe 3 de voorraad notities met orders van klanten geschrapt kan worden en het noteren van de wens van een klant vervangen zou moeten worden door een spoedbestelprocedure. Bij JEEKAA kunnen de verkopers hun taak verbreden door ook spoedbestellingen te doen (toepassing van principe 7), waardoor de tijdrovende overdracht van werk aan de inkoper kan worden vermeden. Eén van de verkopers kan worden opgeleid tot systeembeheerder en zal ook de administratieve controles op de binnengekomen fietsen (kloppen ze met bestelformulier en factuur) voor zijn rekening nemen (dit betekent een kwaliteitsverbetering van de administratie). Met behulp van de computer kan ook het beantwoorden van vragen naar framenummers vereenvoudigd worden.

9. Documentmodellen (D-modellen)

9.1 Documentmodellen (D-modellen)

Documentmodellen (D-modellen, waarbij de D staat voor document) beelden de overdracht en bewaring af van berichten, producten, diensten, geld en rechten. Deze overdracht is de fysieke basis waarop besturing, transformatie en transactie berust. Op grond van een het standaardpatroon van een interactie volgens DEMO kan worden opgesomd uit welke acties een interactie normaliter bestaat, en welke berichten daarmee corresponderen. Daarna kan de communicatie tussen actoren/ systemen worden weergegeven. Hierbij wordt het model gehanteerd van het verzenden van berichten naar postvakjes, en lezen van berichten die in postvakjes (berichtverzamelingen) zijn gelegd. Deze documentmodellen maken gebruik van een deel van de OMT Data Flow notatie.

9.2 Uitwerking interacties in termen van activiteiten en berichten

In het door ons te tekenen documentele model (D-model) komen actoren en berichten voor. Daarnaast worden ook producten, diensten, financiën en rechten die in overdrachtsacties worden overgedragen afgebeeld. Voor het tekenen van een documentmodel is het van belang om eerst de acties op te sommen die door het verzenden en ontvangen van berichten, producten, etc. moeten worden gerealiseerd.

Daartoe maken we een lijst van interacties, interactiefasen, acties en berichten (en andere onderwerpen van acties zoals producten). Hierbij wordt de standaard structuur van de interactie volgens de DEMO theorie gebruikt. Na elke transactie, interactie en actie is tussen haakjes aangegeven welke organisatie de betreffende handeling of reeks handelingen uitvoert. Met elk van de acties correspondeert een bericht. Bij dit bericht is aangegeven van welke organisatie naar welke andere organisatie het bericht gaat. Conditie voor het verzenden van een bericht zijn tussen vierkante haken [] weergegeven. Deze uitwerking is nodig voor de volgende stap: het tekenen van het diagram 'Actoren en Informatie'.

Het is ook verstandig om een dergelijke uitwerking van transacties te maken om de naamgevingsconventies bij het tekenen van diagrammen vast te leggen

Hieronder volgt een uitwerking van de transactie 'Inkoop' van JEEKAA. De namen van de acties die hieronder is weergegeven zijn afgeleid van de standaard naam (verzoek, belofte, etc.) voor de actie. Bij het beschrijven van een bedrijf kan ook de door het bedrijf gehanteerde naam van de actie worden opgenomen met daarachter tussen haakjes de standaard naam. Als een bepaald type bericht niet voorkomt, hoewel het er volgens de standaardindeling wel zou moeten zijn, wordt de naam (geen) vermeld, met eventueel daarachter de reden waarom dit type bericht niet voorkomt.

Transactie:

1. Inkoop (JEEKAA+Leverancier)

Interactie:

- 1.1. Inkoop Levering (Leverancier) (prestatie)
- 1.2. Inkoop Betaling(JEEKAA) (tegenprestatie)

Interactiefase:

- 1.1.1. Opdracht tot leveren fietsen (JEEKAA+Leverancier)
- 1.1.2. Executie van leveren fietsen (Leverancier)
- 1.1.3. Resultaataanvaarding van leveren fietsen (JEEKAA)

- 1.2.1. Opdracht tot betalen fietsen (Leverancier+JEEKAA)
- 1.2.2. Executie van betalen fietsen (JEEKAA)
- 1.2.3. Resultaataanvaarding van betalen fietsen (Leverancier)

Acties:

- 1.1.1.1. Verzoek tot leveren fietsen (JEEKAA)
- 1.1.1.2. Belofte tot leveren fietsen (Leverancier)
- 1.1.2.1. Uitvoering van leveren fietsen (Leverancier)
- 1.1.2.2. Verklaring dat leveren fietsen is uitgevoerd (Leverancier)
- 1.1.3.1. Aanvaarding van verklaring dat leveren fietsen is uitgevoerd (JEEKAA)

- 1.2.1.1. Verzoek tot betalen fietsen (Leverancier)
- 1.2.1.2. Belofte tot betalen fietsen (JEEKAA)
- 1.2.2.1. Uitvoering van betalen fietsen (JEEKAA)
- 1.2.2.2. Verklaring dat betalen fietsen is uitgevoerd (JEEKAA)
- 1.2.3.1. Aanvaarding van verklaring dat betalen fietsen is uitgevoerd (Leverancier)

Onderwerpen van een acties (berichten, producten, diensten, geld):

- 1.1.1.1. Bestelformulier (JEEKAA → Leverancier)
- 1.1.1.2. Orderacceptatie (Leverancier → JEEKAA)
- 1.1.2.1. Fiets (Product) (Leverancier → JEEKAA)
- 1.1.2.2. Vrachtbon (Leverancier → JEEKAA)
- 1.1.3.1. Aftekenen vrachtbon (JEEKAA → Leverancier)

- 1.2.1.1. Inkoopfactuur (Leverancier → JEEKAA)
- 1.2.1.2. (geen; belofte is impliciet gevolg van Bestelformulier + Aftekenen vrachtbon)
- 1.2.2.1. Betaling (F, geld) (JEEKAA → Leverancier)
- 1.2.2.2. [girale betaling] Bank/giroafschrift (JEEKAA → Bank → Leverancier)
- 1.2.3.1. [contante betaling] Kwitantie (Leverancier → JEEKAA)

9.3 Actoren en berichten

Bij het uitvoeren van interacties produceren en gebruiken systemen of actoren informatie in de vorm van berichten. Met elke tot een conversatie behorende actie binnen een interactie correspondeert over het algemeen een bericht. De overdrachtsactie binnen een interactie correspondeert met een product, dienst of hoeveelheid geld die wordt overgedragen. Berichten maken deel uit van een berichtenverzameling. Voor het uitvoeren van een conversatie wordt het volgende model gebruikt: de ene actor schrijft een bericht en legt dit in de berichtenverzameling ('postvakje') en de andere actor neemt dit bericht er uit om het te lezen en legt het bericht daarna eventueel weer terug om het later nog eens te kunnen lezen (dit laatste kan overigens alleen als het bericht geschreven en niet gesproken is).

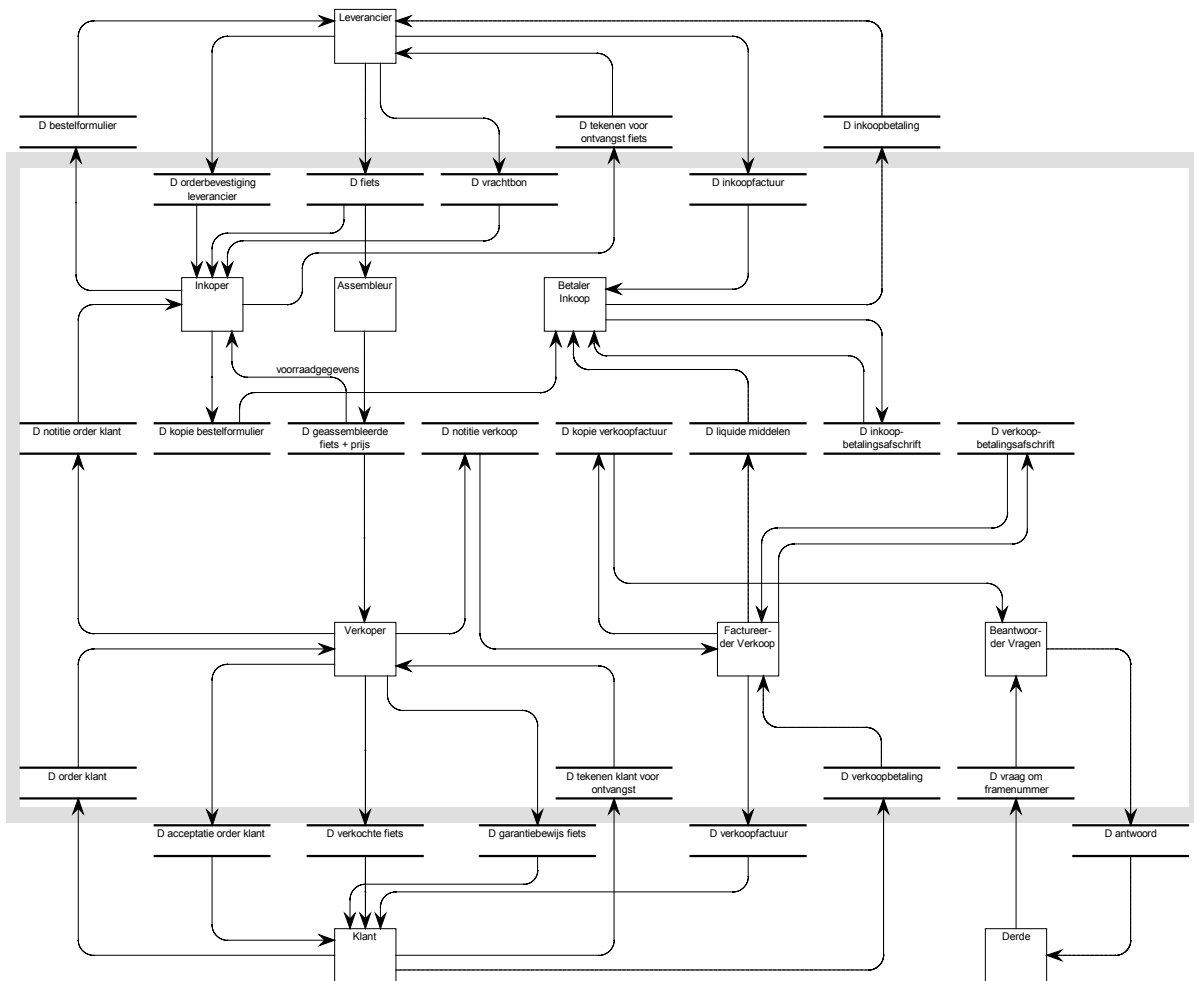
Dit gebeuren wordt weergegeven in het diagram 'Actoren en Berichten' (DATAF). De actoren of systemen worden weer getekend als vierkanten. De berichtverzamelingen ('postvakjes') worden getekend als twee parallelle lijnen (dit lijkt een beetje op een postvakje). Het

schrijven van een bericht en het leggen daarvan in een berichtenverzameling (verzenden) wordt getekend als een pijl naar dit 'postvakje'; het uitnemen, lezen en terugleggen (ontvangen) van een bericht als een pijl vanuit de berichtenverzameling. Omdat dit verzenden en ontvangen van een bericht vele malen kan gebeuren, representeert elke pijl een *verzameling* van verzend- of ontvangacties. Voor het verzenden en ontvangen van producten, diensten of geld worden gestippelde pijlen gebruikt (dit zijn de **control flow** pijlen van OMT die we hiervoor een beetje oneigenlijk gebruiken).

Het diagram 'Actoren en Berichten' (DATAF) wordt met de OMT module Data Flow getekend. Het is geschikt om te zien welke informatie (in termen van soorten berichten) door elke actor worden gebruikt.

Het diagram wordt in drie stappen opgebouwd. Ten eerste worden de actoren en berichtverzamelingen opgenomen die zijn af te leiden uit de geïdentificeerde interacties (zie diagrammen ORGAN en ACTOR) en de standaard structuur van elke interactie. Hierbij kan de lijst die hiervoor net is uitgewerkt (zie paragraaf 7.3.) behulpzaam zijn. Ten tweede wordt het diagram uitgebreid door de interne gegevensverzamelingen als berichtverzamelingen op te nemen. Ten derde wordt gecontroleerd of alle in de verbale beschrijving opgenomen gegevensstromen en gegevensverzamelingen zijn afgedekt, en zo nodig worden de ontbrekende corresponderende berichtverzamelingen opgenomen. Tenslotte wordt gecontroleerd of alle standaardberichten (berichten die er volgens de standaardstructuur van de interactie zouden moeten zijn) ook daadwerkelijk voorkomen.

Het diagram voor JEEKAA is hieronder weergegeven.



We zorgen dat het diagram goed geordend wordt weergegeven. De producten verplaatsen zich van boven naar onder (eventueel van links naar rechts); de financiën verplaatsen zich van onder naar boven. De externe actoren/ organisaties staan bovenaan en onderaan. De berichtverzamelingen zijn in drie lagen gerangschikt: twee lagen voor uitwisseling met de externe actoren en één interne laag. De tijd verloopt van links naar rechts; berichten worden in een logische volgorde van links naar rechts geplaatst. De binnenkomende berichten vallen allemaal binnen de systeemgrens; verzamelingen ervan worden binnen de organisatie bewaard. De uitgaande berichten vallen allemaal buiten de systeemgrens. Als men informatie wil ontlenen aan uitgaande berichten moet men daarvan een kopie bewaren.

9.4 Beoordeling van het informatiegebruik door actoren

Bij de beoordeling van het informatiegebruik bij de uitvoering van activiteiten door actoren moet met name gekeken worden naar (1) ondoelmatige wijzen van het verkrijgen van informatie (zoals bijvoorbeeld het tellen van fysieke voorraden) en (2) de kwaliteit van de gebruikte beslisregels. Het uitvoeren van taken kan vaak beter als beslisregels worden gehanteerd die bewezen hebben goed te werken en daarom in handboeken e.d. zijn vermeld.

Vanuit het oogpunt van doelmatigheid bij het verkrijgen van informatie is het onwenselijk om steeds weer de fysieke verzameling fietsen te moeten tellen. Het is gewenst dat de verzameling voorradige fietsen in de computer is vastgelegd om het tellen te vermijden. Dit betekent dat het verkopen van fietsen en het controleren van gearriveerde fietsen anders zal

gaan omdat het een en ander met behulp van de computer moet worden geregistreerd. Deze registratie zal plaatsvinden met behulp van een te realiseren *fietsinformatiesysteem*.

Op het gebied van beslisregels is het bij JEEKAA zinvol om de activiteit *bestellen fietsen* die vooraf gaat aan de actie 'verzoek tot leveren fietsen' en het bericht 'bestelformulier' verder te analyseren. Deze analyse luidt als volgt. De inkoper maakt gebruik van de stapel notities met orders van klanten en de fysieke voorraad fietsen om na te gaan wat besteld moet worden. Het tellen van de aanwezige fietsen is tijdrovend. Uit een vervolginterview met de inkoper blijkt dat hij bestelt wat door de klanten is besteld (op grond van notities) en wat volgens hem ontbreekt in het standaard-assortiment (op grond van het opnemen van de voorraad). Zijn formule voor het inkopen luidt als volgt:

$$I = F + (A - V)$$

I: verzameling in te kopen fietsen

F: verzameling door klanten bestelde fietsen die niet in voorraad is

A: verzameling fietsen die geldt als basisassortiment voor dit kwartaal (dames en herenfiets van de meest verkochte typen)

V: verzameling fietsen in voorraad

Het is echter weinig rationeel om met een niet expliciet gemaakt basisassortiment te werken. Het werken met een basisassortiment moet expliciet worden gemaakt en door de computer worden ondersteund. Hierbij kan bijvoorbeeld per kwartaal een basisassortiment worden vastgesteld n.a.v. de catalogi van de leveranciers en de voorspelde vraag. Wekelijks zou dit basisassortiment kunnen worden aangevuld. Voor fietsen en accessoires die goed lopen kan met behulp van de bestelformule wekelijks een optimale bestelhoeveelheid worden bepaald.

10. Objectmodellen (I-modellen)

10.1 Objectmodellen (I-modellen)

Objectmodellen (I-modellen, de I staat voor informatie) zijn modellen van de mogelijke uitspraken die door een informatiesysteem kunnen worden vastgelegd. De objectmodellen die wij behandelen zien de mogelijke uitspraken die kunnen worden gedaan over de wereld in termen van objecttypen en associatietypen. De uitspraken die op grond van het onderscheiden van een objecttype kunnen worden gedaan zijn van de aard:

Het object van objecttype A met naam B heeft een kenmerk C met waarde D.

Voorbeeld:

We onderscheiden het objecttype AUTO. We kunnen nu de volgende informatie vastleggen:

De auto met kenteken SP-04-DL heeft de kleur rood.

A = AUTO; B = SP-04-DL; C = KLEUR; D = rood.

De uitspraken die op grond van het onderscheiden van een associatietype kunnen worden gedaan zijn van de aard:

Het object van objecttype A met naam B heeft een associatie genaamd C met een object van objecttype D genaamd E.

Voorbeeld:

We onderscheiden het associatietype IS UITGELEEND AAN tussen de objecttypen CD en PERSOON. We kunnen nu de volgende informatie vastleggen:

De CD genaamd 'Irish Heartbeat' is uitgeleend aan de persoon Karel Hageman.

A = CD; B = Irish Heartbeat; C = IS UITGELEEND AAN; D = PERSOON; E = Karel Hageman.

Nadat objecttypen en associatietypen zijn vastgesteld moeten de kenmerken (attribuuttypen) ervan nog worden gespecificeerd.

Het vaststellen van objecttypen en associatietypen is belangrijk omdat het over de kern van een informatiesysteem gaat, namelijk het model van de wereld dat het informatiesysteem kent. Zonder objecttypen en associatietypen kan geen informatiesysteem bestaan. Voor het maken van een dergelijk model zijn er twee invalshoeken: een wiskundige en een filosofische. Er is gekozen voor de meer filosofisch getinte methode, te weten objectgeoriënteerde modellering. Een eerste versie van het objectmodel kan met behulp van enkele regels 'afgeleid' worden uit de eerder gemaakte modellen, en vervolgens verfijnd. Deze beschrijving maakt gebruik van de OMT Object Model notatie.

10.2 Objecttypen

Bij het maken van onze lijst met objecttypen hanteren we de reeks categorieën *actor*, *transactie*, *verzameling die onderwerp is van transactie*, *individu*, *type* als leidraad.

De voor ons relevante *actoren* en *transacties* kunnen worden afgeleid uit het eerste actor-interactie-diagram (ORGAN). Voor JEEKAA krijgen we de volgende objecttypen die met actoren (in dit geval organisaties) corresponderen:

- JEEKAA;
- KLANT;
- LEVERANCIER;
- DERDE.

De volgende objecttypen corresponderen met *transacties* (objecttypen zijn altijd enkelvoud; dit vergt herschrijven van de namen van de transacties):

- INKOOPTANSACTIE;
- VERKOOPTRANSACTIE;
- BEANTWOORDING VRAAG.

Eigenlijk moeten we voor elke interactie één objecttype onderscheiden. Als een transactie echter homogeen is, dat wil zeggen dat alle fasen in een transactie op dezelfde verzameling producten, diensten, etc. betrekking heeft, mogen we deze objecttypen echter samenvoegen tot één objecttype voor de hele transactie. Dat is wat hier is gebeurd. Immers, we gaan ervan uit dat zowel levering als betaling op dezelfde verzameling fietsen betrekking hebben. Een levering wordt bijvoorbeeld niet in twee keer betaald.

Vervolgens komt een moeilijke stap: we moeten nagaan welke *verzamelingen* onderwerp van deze transacties zijn. Het gaat in alle gevallen om verzamelingen fietsen. Bij een INKOOPTANSACTIE horen enkele verzamelingen INGEKOCHTE FIETSEN. Idem dito horen bij een VERKOOPTRANSACTIE in principe ook enkele verzamelingen VERKOCHTE FIETSEN. Voor het gemak stellen we dat het onderwerp van BEANTWOORDING VRAAG een verzameling fietsen is die uit één fiets bestaat: NAGEVRAAGDE FIETS.

De hierna volgende stap is kijken uit welke *individuen* de onderscheiden verzamelingen bestaan. Het gaat hier steeds om individuen van het objecttype FIETS.

Tenslotte kijken we naar de *types* waartoe die individuen behoren. We hoeven alleen maar het objecttype FIETSTYPE te onderkennen.

Als we alles samenvatten en ordenen, krijgen we in het geval van JEEKAA de volgende naar categorieën ingedeelde lijst met objecttypen:

actoren:

- JEEKAA;
- KLANT;
- LEVERANCIER;
- DERDE.

transacties (in min of meer chronologische volgorde):

- INKOOPTANSACTIE;
- VERKOOPTRANSACTIE

- BEANTWOORDING VRAAG;
- verzamelingen:*
- INGEKOCHTE FIETSEN;
- VERKOCHTE FIETSEN;
- NAGEVRAAGDE FIETS;
- individuen:*
- FIETS.
- typen:*
- FIETSTYPE.

10.3 Associatietypen

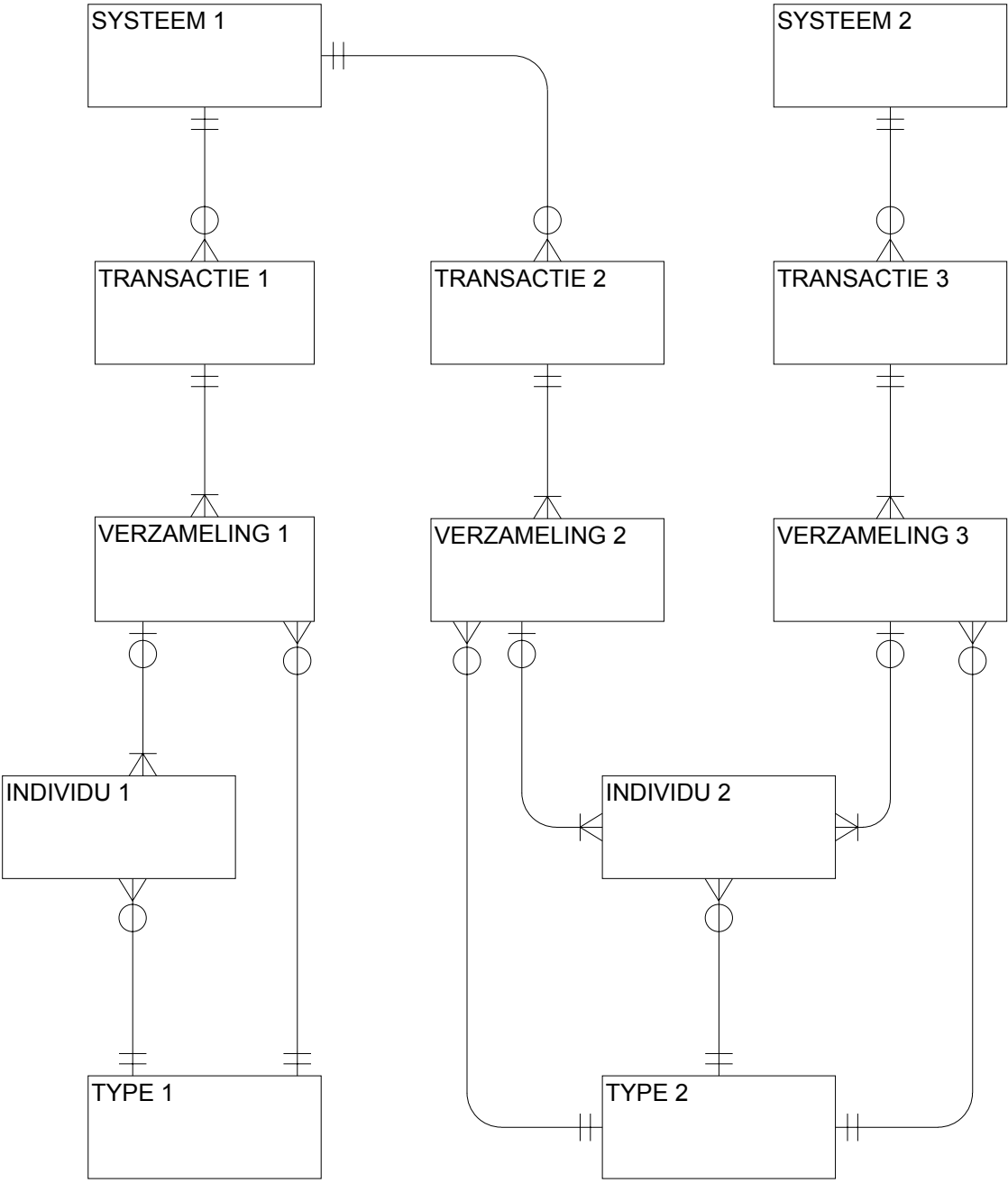
Bij het bepalen van de associatietypen gaan we uit van de volgende proposities (erachter is de eruit volgende multipliciteitsregel aangegeven):

Nr.	Propositie	A	B	A	A	B	B
				>	<	>	<
				=	=	=	=
1.	een actor S kan bij al dan niet bij transacties T zijn betrokken	S	T			0	M
2.	bij een transactie T is altijd één actor S betrokken;	S	T	1	1		
3.	een transactie T omvat verzamelingen V;	T	V				M
4.	een transactie T omvat tenminste één verzameling V;	T	V			1	
5.	een verzameling V hoort altijd bij één bepaalde transactie T;	T	V	1	1		
6.	een verzameling V kan bestaat uit individuen I;	V	I				M
7.	een verzameling bestaat uit tenminste één individu					1	
8.	een individu kan, maar hoeft niet, lid te zijn van een verzameling V;	V	I	0	1		
9.	een verzameling V is altijd van één type T;	T	V	1	1		
10.	een type T hoeft niet als verzameling V voor te komen	T	V			0	
11.	een type T is toepasbaar op meerdere individuen I	T	V				M
12.	een individu I is altijd van één type T;	T	I	1	1		
13.	een type T hoeft niet als individu I voor te komen.	T	I			0	
14.	een type T is toepasbaar op meerdere verzamelingen	T	I				M

Legenda	
A	de letter A slaat op het objecttype dat wordt aangeduid met letter:
B	de letter B slaat op het objecttype dat wordt aangeduid met letter:
A>=	de cardinaliteit van A is minimaal:
A<=	de cardinaliteit van A is maximaal:
B>=	de cardinaliteit van B is minimaal:

B<=	de cardinaliteit van B is maximaal:
-----	-------------------------------------

Op grond van deze algemene proposities zou een objectmodel er bijvoorbeeld uitzien zoals hieronder is weergegeven.



Bij deze meer algemene beweringen kunnen we eventueel extra proposities toevoegen voor de te modelleren organisatie. Bij JEEKAA is dat:

- 15. een FIETS behoort altijd tot de eens INGEKOCHTE FIETSEN (buiten de inkoop om kan er geen fiets bij JEEKAA terecht komen).

Deze propositie schakelt propositie 8 uit voor het associatietype INGEKOCHTE FIETSEN-FIETS en zegt dat de minimale en maximale cardinaliteit van INGEKOCHTE FIETSEN in de multipliciteitsregel voor het associatietype INGEKOCHTE FIETSEN-FIETS één is.

16. de verzameling NAGEVRAAGDE FIETS bestaat altijd uit één FIETS.

Deze bewering schakelt propositie 6 uit voor het associatietype NAGEVRAAGDE FIETS - FIETS en zegt dat de maximale cardinaliteit van NAGEVRAAGDE FIETS in de multipliciteitsregel voor het associatietype NAGEVRAAGDE FIETS -FIETS één is.

10.4 Objectmodel

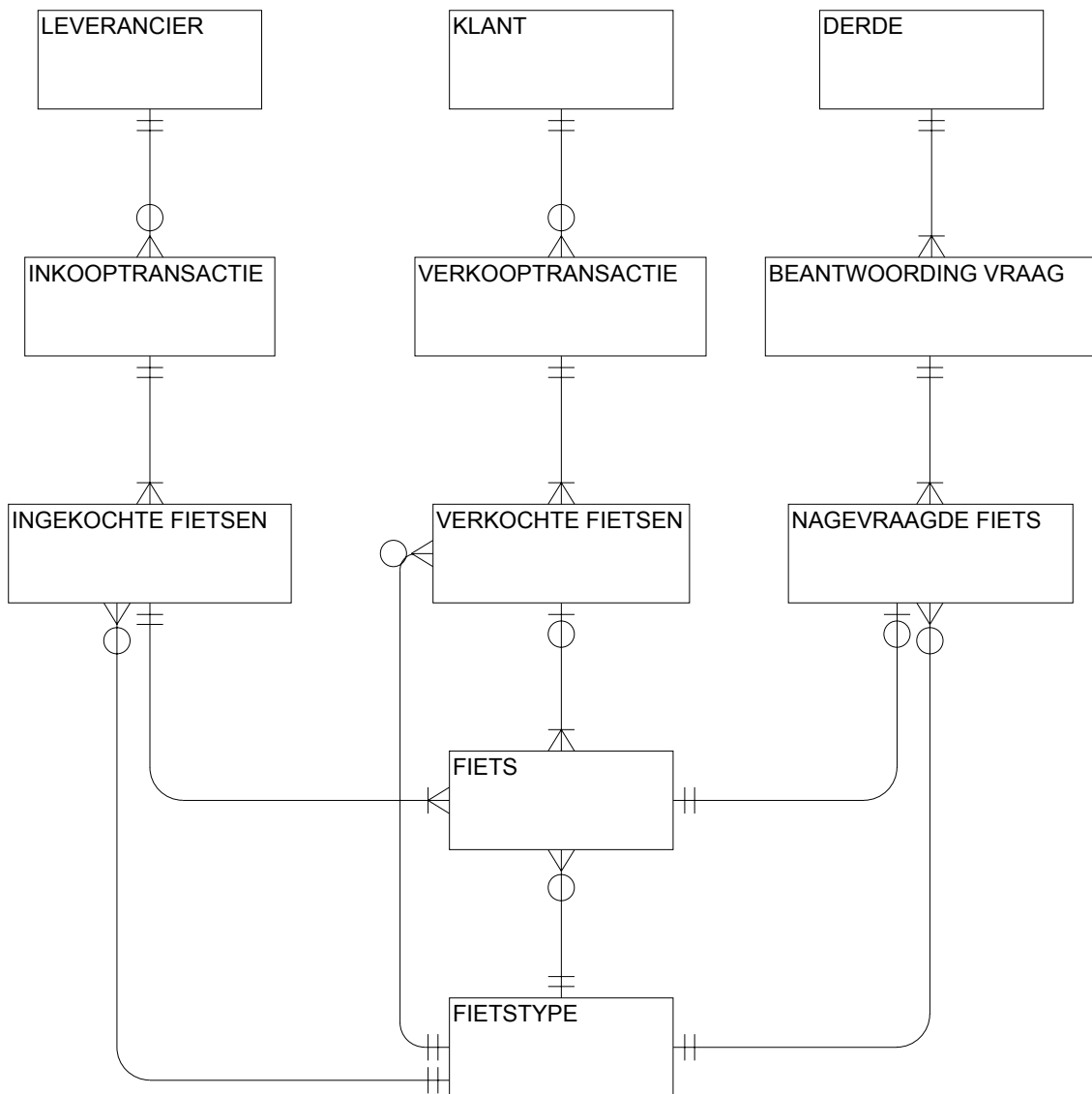
Nu kan het objectmodel (naam: OBJEC) worden getekend. Dit kan met System Architect module OMT Object Model. De organisaties en transacties kunnen we kopiëren uit ons ORGAN diagram en daarna door middel van <rechtermuisklik>| **Transform Symbol** omzetten naar objecttypen (**class** in OMT Object model). Voor het tekenen van de associatietypen gebruiken we de **association** lijn (let op: niet de **generalization line** gebruiken). Bij **Set|Notation** kiezen we **Crow's Feet** om nette kraaiepotjes te krijgen.

Het is gebruikelijk om in het objectmodel het objecttype dat betrekking heeft op het bedrijf zelf (bij ons: JEEKAA) weg te laten.

De objecttypen worden in nette rijen gerangschikt, die, van boven naar beneden, bestaan uit:

- actoren;
- transacties;
- verzamelingen;
- individuen;
- typen.

We krijgen zo het volgende objectmodel (naam: JKOBJEC1).



Dit objectmodel is eenvoudig uit te bereiden met attribuuftypen. Dat zijn de kenmerken die we moeten registreren van de objecttypen om uitspraken vast te leggen. Bijvoorbeeld als we de uitspraak:

"De afdeling Inkoop van JEEKAA heeft Bestelling #15 op 3 maart 1997 geplaatst bij Gazelle."

willen registreren moeten we een objecttype INKOOPTRANSACTIE hebben met daarin de attribuuftypen:

- organisatie ("Afdeling Inkoop van JEEKAA");
- bestelnr ("Bestelling #15");
- besteldatum ("3 maart 1997");
- leveranciersnr (nr 1, behorende bij LEVERANCIER "Gazelle").

Hierbij is bestelnr de sleutel van de INKOOPTRANSACTIE, d.w.z. het attribuuftype dat elke afzonderlijke inkooptransactie uniek identificeert. Elk objecttype dient een *sleutel* te hebben, een uniek identificerend attribuut. Om deze reden worden vaak nummers ingevoerd, zoals

bijvoorbeeld het leveranciersnr dat een leverancier uniek identificeert. Dit leveranciersnr gebruiken we om in het objecttype INKOOPTTRANSACTIE aan te duiden bij welke LEVERANCIER een bepaalde inkooptransactie hoort. Een attribuut dat op deze manier wordt gebruikt om naar een ander objecttype te verwijzen heet een *verwijzende sleutel*. All associaties in het objectmodel moeten door verwijzende sleutels worden vormgegeven. Daarbij wordt altijd de verwijzende sleutel opgenomen in het objecttype waarbij het kraaietpootje (het méér teken) staat.

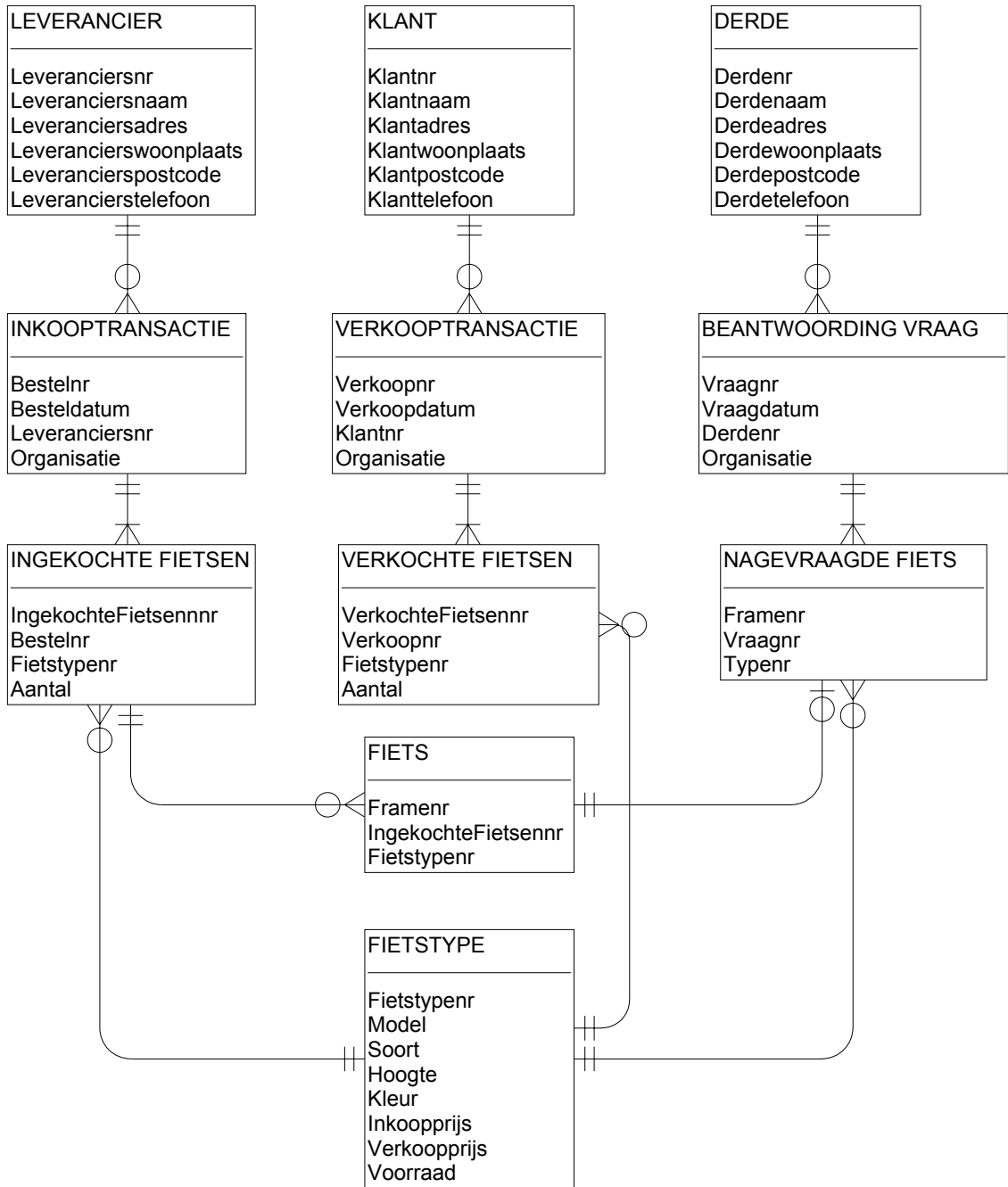
Voorbeeld:

Het objecttype VERKOCHTE FIETSEN dient een verwijzende sleutel verkoopnr te hebben die verwijst naar de sleutel verkoopnr van het objecttype VERKOOPTTRANSACTIE. Op deze manier wordt de 1:M-associatie tussen VERKOOPTTRANSACTIE en VERKOCHTE FIETSEN vormgegeven.

Het tekenen van de attribuuttypen gaat in System Architect als volgt. Met **<rechtermuisklik>** | **Definition** wordt een venster voor de definitie van Attributes en Operations opgeroepen. Hierna kan naast de knop **New** de naam van het attribuuttype worden ingevuld. Me de knop **Add** wordt dit attribuuttype aan de lijst toegevoegd. Als de lijst klaar is op OK klikken. Men krijgt dan de boodschap 'Undefined Class Attribute . . .' die genegeerd kan worden; hierbij op **No** klikken.

Een uitgewerkt objectmodel voor JEEKAA met daarin ook de attribuuttypen is hieronder weergegeven (naam van dit diagram: JKOBJEC2).

We zien dat de attribuuttypen van de verschillende actoren een zekere gelijkenis vertonen, evenals die van de transacties en de onderwerpen van de transacties.



11. Case Fietsenhandel JEEKAA

11.1 Omschrijving

Fietsenhandel JEEKAA, opgericht door Jeen Kaats, bestaat al 22 jaar. De laatste jaren is het bedrijf door toenemende vraag gestaag gegroeid. Het is een typische eenmanszaak. Er werken naast de eigenaar nog twee verkopers en drie onderhoudsmonteurs. Voor de koopavond en de zaterdag is er nog een parttime verkoper. De laatste tijd heeft de eigenaar het gevoel dat hij het overzicht wat verliest. Om zijn voorraadadministratie doorzichtelijker te maken verzoekt de eigenaar U een oplossing te zoeken. Hieronder staan enkele details van de gang van zaken in het bedrijf.

De activiteiten die betrekking hebben op de voorraad vallen uiteen in drie categorieën. Het bestellen van de fietsen, het arriveren van bestelde fietsen en het verkopen van fietsen. Deze drie processen worden nu handmatig uitgevoerd.

Het bestellen van fietsen gebeurt met behulp van bestelformulieren (zie bijlage 1). Het bestelmoment is van een aantal factoren afhankelijk. Over het algemeen worden de bestelformulieren ingevuld op een moment waarop de eigenaar toevallig even wat tijd heeft. De bestellingen gebeuren grotendeels op het gevoel. Er bestaat geen duidelijk inzicht in de aanwezige voorraad. Soms vraagt een klant om een specifieke fiets die niet in voorraad is. Deze fiets wordt dan op dat moment genoteerd om besteld te worden. Kopieën van de bestelformulieren worden bewaard in een ordner.

Als bestelde fietsen binnenkomen, wordt hiervan de vrachtbon bewaard in een ordner. De factuur, die bij aflevering wordt meegegeven of later wordt opgestuurd, wordt ook in een ordner opgeborgen. Bij binnenkomst van een bestelde fiets moet in principe hiervan een aantekening op het bijbehorende bestelformulier komen. In de praktijk gebeurt dit echter nooit. Het terugzoeken van de juiste bestelformulier is tijdrovend, en bovendien vinden de werknemers het zinloos. Na het binnenkomen van de fiets wordt deze door een technicus uitgepakt en gecontroleerd. Vervolgens worden, indien de klant dat heeft besteld, bepaalde speciale onderdelen zoals andere zadels, bagagedragers, drinkflessen, fietstassen, sloten gemonteerd.

Bij verkoop van een fiets worden de gegevens van de fiets en de klantgegevens handmatig ingevuld op een voorgedrukt formulier. Dit formulier dient als verkoopfactuur. Een doorslag wordt gebruikt voor de administratie van de winkel. Een tweede doorslag van dit formulier wordt eventueel voor de verzekering gebruikt. Deze verzekering betreft voor brommers de WA-verzekering en voor fietsen en brommers een diefstalverzekering. Nadat een fiets is verkocht wordt de fiets door een technicus voor aflevering klaar gemaakt door onder meer schoonmaken, in de was zetten, oliën en banden oppompen.

Het gebeurt steeds vaker dat er gegevens moeten worden opgezocht. Het komt bijvoorbeeld voor dat een bestelling geannuleerd wordt. Dit moet op het bijbehorende bestelformulier worden vermeld. Het terugvinden van dit formulier is lastig. Er is geen overzicht over de aanwezige voorraad. Het komt regelmatig voor dat er om een specifieke fiets wordt gevraagd. De voorraad is echter gegroeid tot zo'n 400 fietsen, waardoor het moeilijk is om een specifieke fiets terug te vinden. Hier komt nog bij dat de eigenaar nu niet precies weet wat hij moet bestellen. Dit gebeurt nu min of meer intuïtief. Wanneer er vragen binnenkomen over

een verkochte fiets zijn de gegevens van deze fiets moeilijk terug te vinden. Dit gebeurt vooral in het geval van diefstal: een klant wil het framenummer van zijn fiets weten, of de politie wil van een gevonden fiets de eigenaar achterhalen.

11.2 Bestelformulier

BESTELFORMULIER				
Volgnummer	<input style="width: 100%;" type="text"/>			
JEEKAA Tweewielers Kooistraat 12 3354 FF Schiermonnikoog Kamer van Koophandel Groningen 3251				
Datum	<input style="width: 100%;" type="text"/>			
Aan:	Leveranciersnr:	<input style="width: 100%;" type="text"/>		
Naam	<input style="width: 100%;" type="text"/>			
Adres	<input style="width: 100%;" type="text"/>			
Postcode	<input style="width: 100%;" type="text"/>			
Woonplaats	<input style="width: 100%;" type="text"/>			
Hierbij plaats ik de volgende bestelling:				
Type	Soort	Hoogte	Kleur	Aantal
Hoogachtend, Jeen Kaats, directeur				

11.3 *Factuur*

FACTUUR						
Factnr	<input style="width: 100%;" type="text"/>					
Batalle Fietsenfabriek						
Groeneweg 27						
8233 AF Beesten						
Datum	<input style="width: 100%;" type="text"/>					
Klantnr	<input style="width: 100%;" type="text" value="441"/>					
Naam	JEEKAA					
Adres	Kooistraat 12					
Postcode	NL-3354 FF					
Woonplaats	Schiermonnikoog					
Type	Soort	Hoogte	Kleur	Framenr	Prijs	Korting
Totaal						
BTW 17,5%						
Factuurbedrag						
Betaling binnen 14 dagen. Kamer van Koophandel Peerstekel nr.3456						

12. Literatuur

- Algra, N.E., J.B.J.M. ten Berge en P.H. Sleurink. Poly-Juridisch Zakboekje. Arnhem: PBNA, 1986.
- Bekke, J.H. ter. Database ontwerp: Tweede, geheel herziene druk. Leiden: Stenfert Kroese, 1988.
- Boertjens, Koos. Basiscursus Access 7 voor Windows 95: UK versie. Schoonhoven: Academic Service, 1996.
- Carley, K.M., and M.J.Prietula (eds.). Computational Organization Theory. Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.
- Dietz, J.L.G. Introductie tot DEMO: Van informatietechnologie naar organisatietechnologie. Alphen a/d Rijn: Samsom, 1996.
- Dietz, J.L.G. Leerboek Informatiekundige Analyse. Deventer: Kluwer Bedrijfswetenschappen, 1992.
- Frissen, P.H.A. De virtuele staat: Politiek, bestuur, technologie: Een postmodern verhaal. Schoonhoven: Academic Service, 1996.
- Gale, Thornton, and James Eldred. Getting Results with the Object-Oriented Enterprise Model. New York: SIGS, 1996.
- Gazendam Henk W.M. Variety Controls Variety: On the Use of Organization Theories in Information Management. Groningen: Wolters-Noordhoff, 1993.
- Gazendam, H.W.M., en S.Sibum. Bouw van Informatiesystemen. Groningen: Faculteit Bedrijfskunde, 1995.
- Gazendam, H.W.M., en D.J.Schaap. Het gebruik van Feiten en Cijfers: Een onderzoek naar de rol van bestuurlijke informatie over hoger onderwijs en wetenschappelijk onderzoek: Achtergrondstudies Hoger onderwijs en Wetenschappelijk onderzoek 22. Zoetermeer: Ministerie van Onderwijs, Cultuur en Wetenschappen, 14 oktober 1994.
- Gazendam, Henk W.M. "Object modeling of databases and multi-actor systems: A future-oriented report for ENCOMPASS.", *Projectrapport*, Groningen: IKS, 1994.
- Gazendam, Henk W.M. (projectleider). Besturing Informatievoorziening Onderwijs en Wetenschappen: Visie 1985-1990. Zoetermeer: Ministerie van Onderwijs en Wetenschappen, 31 juli 1985.
- Gazendam, Henk W.M., and René J.Jorna. "Theories about Architecture and Performance of Multi-Agent Systems." *Paper presented at the III European Congress of Psychology, July 4-9-1993, Tampere, Finland.*
- Gifford, Dwayne, e.a. Access 95 Unleashed. Indianapolis, IN: SAMS, 1996.
- Hammer, Michael. "Reengineering Work: Don't Automate, Obliterate.", *Harvard Business Review*, July-August 1990, pp.104-112.
- Jacobson, Ivar, e.a. Object-Oriented Software Engineering: A Use Case Driven Approach. Wokingham: Addison-Wesley, 1992.
- Jorna, R.J., H.W.M.Gazendam, H.C.Heesen en W.M.C.van Wezel. Plannen en Roosteren: Taakgericht analyseren, ontwerpen en ondersteunen. Leiderdorp: Lansa, 1996.
- Kranenburg, Kees, en Ad van Riel. Model-based Application Development (MAD): .Deventer: Kluwer Bedrijfswetenschappen, 1995.
- Laudon, Kenneth C., and Jane Price Laudon. *Management Information Systems: Organization and Technology: Fourth Edition.* Upper Saddle River, NJ: Prentice-Hall, 1996.

- Starreveld, R.W., e.a. Bestuurlijke Informatieverzorging. Alphen a/d Rijn: Samsom, 1981.
- Leeuw, A.C.J. de. *Organisaties: Management, analyse, ontwerp en verandering: 3e druk*. Assen: Van Gorcum, 1988.
- Newell, A., and H.A.Simon. Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall, 1972.
- Ould, Martyn A. Business Processes: Modelling and Analysis for Re-engineering and Improvement. Chichester: Wiley, 1995.
- Peirce, Charles Sanders. *Selected Writings: Values in a Universe of Chance*. Philip P. Wiener (ed.). New York: Dover, 1958.
- Rumbaugh, James, Michael Blaha, William Premerlani, Frederick Eddy and William Lorensen. Object-Oriented Modeling and Design. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- Russell, Stuart J., and Peter Norvig. Artificial Intelligence: A Modern Approach. Upper Saddle River, NJ: Prentice-Hall, 1995.
- Schneiderman, Ben. Designing the User Interface. Reading, MA: Addison-Wesley, 1987.
- Searle, John R. Speech Acts. 1969. Vertaald als: Taalhandelingen. Utrecht: Het Spectrum, 1977.
- Structuurschetsen voor de Interbestuurlijke Informatievoorziening: Bestuurlijke Overlegcommissie voor Overheidsautomatisering: Rapport nr. 12. 's-Gravenhage: Staatsuitgeverij, 1983.
- Terug naar de toekomst: Over het gebruik van informatie en informatie- en communicatietechnologie in de openbare sector: Beleidsnota Informatiebeleid Openbare Sector nr. 3 (BIOS-3). Den Haag: Ministerie van Binnenlandse Zaken, juli 1995.
- Williamson, Oliver E. The Economic Institutions of Capitalism. New York: The Free Press, 1985.

13. Bijlage 1. Typologie Starreveld