

READER

behorende bij het college

a136A INFORMATIESYSTEMEN

Prof. dr. H.G. Sol

1988

READER

behorende bij het college

al36A Informatiesystemen

Prof. dr. H.G. Sol

1987

Inhoud

1. A. Bosman en H.G. Sol , Bestuurlijke Informatiekunde in het Wetenschappelijk Onderwijs, Informatie, 25, nr. 6, 1983.
2. H.G. Sol , Kennis en Ervaring rond het Ontwerpen van Informatiesystemen, Informatie, 27, nr. 3, 1985.
3. J.J. van Griethuysen en D.A. Jardine , De Infomod-benadering van Informatiemodellering, Informatie, 26, nr. 6, 1984.
4. I.G. McDonald , Information Engineering, James Martin Associates, 1986.
5. P.G.W. Keen , Information Systems and Organizational Change, Communications of the ACM, 24, nr. 1, 1981.
6. S.M. Argelo , Valkuilen bij Automatieprojekten, NGI-rapport 5A.
7. A. Kranendonk , Een socio-technische aanpak: automatisering voor de rechter hersenhelft.
8. H.G. Sol , De gebruiker in model.

BESTUURLIJKE INFORMATIEKUNDE IN HET WETENSCHAPPELIJK ONDERWIJS

door prof. dr. A. Bosman en dr. H. G. Sol

Dit artikel gaat in op de vraag hoe toepassingen van de informatica moeten worden georganiseerd in het wetenschappelijk onderwijs en aan welke problemen van inhoudelijke aard bij de constructie van een curriculum aandacht moet worden besteed.

1 INLEIDING

Over de toepassingen van informatica wordt in Nederland uitvoerig gediscussieerd, maar er wordt weinig ondernomen om te komen tot curricula waarin toepassingen van de informatica een leidende rol spelen. In dit artikel zullen we in par. 2 een schets geven van de bestaande situatie op één gebied van toepassing, nl. dat van het gebruik van informatica t.b.v. het besturen van organisaties. Dit is ongetwijfeld momenteel het grootste gebied van toepassing, echter zeker niet het enige.

Veel van de problemen die zich voordoen bij dit toepassingsgebied, zullen ongetwijfeld optreden of gaan optreden bij andere toepassingen. Ons verhaal is er op gericht anderen enige informatie te geven over wat ze te wachten staat. Wellicht kunnen ze er van leren. Er zijn twee problemen die bij elke toepassing centraal zullen staan.

Het eerste betreft de wijze waarop de toepassing onderwijs technisch en organisatorisch een inhoud krijgt. Wij zijn van mening, en we lichten dat toe in par. 3, dat dit in het Nederlandse systeem alleen kan geschieden via het creëren van een specialisatie, in een tweefasenstructuur bij voorkeur door het instellen van een studierichting. Het tweede probleem is de inhoud die aan een specialisatie moet worden gegeven. In par. 4 zetten wij uiteen dat dit voor de problematiek van het besturen van organisaties alleen kan via het specificeren van een rode draad tussen de vakken in een curriculum. In ons geval wordt die rode draad gevonden in een combinatie van een paradigma in het object van onderzoek en methoden van onderzoek. Wij zijn van mening dat ook bij andere gebieden van toepassing een oplossing in die richting moet worden gezocht. Dit behoeft overigens niet te betekenen dat er maar één rode draad zou kunnen worden onderscheiden. In par. 5 beschrijven we een curriculum aan de hand van onze rode draad en de vakkenspecificatie in de aanvraag voor een studierichting bestuurlijke informatiekunde van de sectie economie van de Academische Raad. We sluiten dit artikel af met in par. 6 een vergelijking te maken met het buitenland. In die vergelijking wordt op een tweetal punten die in de voorafgaande paragrafen centraal hebben gestaan verder ingegaan. Voorts wordt daarbij geconstateerd dat Nederland een achterstand op het gebied van de breedte van het onderwijs binnen universiteiten t.o.v. het buitenland vertoont. Alhoewel we op dit punt niet ingaan, is het onze hypothese, dat die achterstand innovatieprocessen in ons land op het gebied van gebruik van micro-elektronica niet van de grond doet komen. Een blokkering die op de lange duur ernstige gevolgen kan hebben voor de economische en maatschappelijke ontwikkeling.

2 DE BESTAANDE SITUATIE IN HET WETENSCHAPPELIJK ONDERWIJS

De studierichting informatica is in 1981 van start gegaan bij een aantal instellingen van wetenschappelijk onderwijs (w.o.). Het curriculum van deze studierichting voorziet in de mogelijkheid afstudeerrichtingen te specificeren. Een aantal van deze afstudeerrichtingen kan zijn gericht op bepaalde toepassingen, waaronder die van het gebruik van computers t.b.v. het besturen van organisaties. Daarnaast is het mogelijk dat binnen andere studierichtingen, bijv. die van de economie en de bedrijfskunde, afstudeerrichtingen worden gedefinieerd waarbij veel aandacht wordt geschonken aan de informatica. Deze situatie kan aanleiding zijn tot het ontstaan van misstanden en misverstanden. De belangrijkste redenen daarvoor zijn de volgende:

1. Een afstudeerrichting heeft binnen het Academisch Statuut geen formele status. Dat betekent dat elke combinatie van vakken met bijv. het vak informatica als een afstudeerrichting kan worden omschreven. De inhoud van een afstudeerrichting en de omschrijving ervan kan van instelling tot instelling verschillen[1]. Wanneer het gaat om de ontwikkeling van nieuwe vakgebieden kan een dergelijke onduidelijke situatie leiden tot misverstanden in het maatschappelijke verkeer.
2. In vele studierichtingen is of wordt een inleiding in de informatica een verplicht onderdeel van het curriculum. Dit feit moet op zich zelf worden toegejuicht. Het geeft echter weinig of geen garantie dat de ontwikkeling van de toepassingen van informatica sterk zal worden gestimuleerd. Voor zover daarvoor garanties kunnen worden gegeven, zullen die vooral afhangen van de wijze waarop de studierichtingen, die zich met de toepassing bezig houden, in staat zijn de informatica te integreren. Wij zijn van mening dat de suggestie dat stimulering van het onderwijs in de informatica op korte of zelfs lange termijn zou leiden tot een groei en/of verbetering van de toepassing op grote schaal berust op het misverstand dat deze integratie gemakkelijk realiseerbaar is. Dit misverstand leidt tot de misstand dat, zeker in Nederland, veel te weinig aandacht wordt geschonken aan het gebied van de toepassingen van de informatica.
3. Informatica kan in vrijwel alle studierichtingen worden toegepast. Vooral in de empirische wetenschappen kan een integratie van de informatica leiden tot een aanzienlijke verandering in de wijze waarop het betrokken vakgebied wordt bestudeerd.

Dergelijke veranderingen vereisen veel tijd.

De drie genoemde redenen hebben de sectie economie van de Academische Raad (A.R.) ertoe gebracht in juni 1982 een voorstel bij de A.R. in te dienen tot het instellen van een studierichting bestuurlijke informatiekunde (b.i.k.). In overleg met de sectie informatica van de A.R. (ARSI) wordt door de sectie economie voorgesteld de afstudeerrichting, die zich in het bijzonder bezig houdt met problemen rond de introductie en het gebruik van computers in organisaties, binnen de studierichting informatica te omschrijven als bestuurlijke informatica (b.i.). Dit voorstel sluit aan bij de bestaande toestand binnen het HBO. De b.i.k. komt overeen met de bedrijfsinformatica-opleiding (BIO.) binnen het HEAO; de studierichting informatica met de hogere informatica opleiding (HIO.) binnen het HTO. In de HIO wordt de nodige aandacht besteed aan de b.i. Op dit moment (eind maart 1983) is de situatie als volgt:

- de studierichting informatica is bij een aantal instellingen begonnen in het cursusjaar 1981 - 1982;
- de A.R. heeft het voorstel van de sectie economie tot de instelling van een studierichting b.i.k. overgenomen.

De minister heeft het voorstel tot het instellen van die studierichting in behandeling. Bij die behandeling moet rekening worden gehouden met de volgende feiten:

- Het college van bestuur van de Katholieke Hogeschool Tilburg (K.H.T.) heeft begin 1982 een aanvraag ingediend voor het kunnen instellen van een experimentele studierichting b.i.k. binnen de economische faculteit van de Hogeschool. De adviesorganen van de A.R., waaronder de secties informatica en economie, hebben geadviseerd het voorstel van de sectie economie te steunen en daarmee het voorstel tot het instellen van een experimentele studierichting af te wijzen. Volgens de laatste berichten, een brief van de minister van O. en W. d.d. 22 februari aan de colleges van besturen, heeft de minister de aanvraag voor het instellen van een experimentele studierichting b.i.k. aan de K.H.T. wel gehonoreerd. Dat laatste betekent dat de andere economische faculteiten voor tenminste vijf jaren formeel niet tot de instelling van een b.i.k. kunnen overgaan.
- Het college van bestuur van de Erasmus Universiteit te Rotterdam (E.U.R.) heeft in 1982 een voorstel ingediend voor het instellen van een studierichting b.i. aan de economische faculteit van die universiteit. Dit voorstel is o.a. een gevolg van het feit dat de E.U.R. niet beschikt over een faculteit wiskunde en natuurkunde en derhalve ook niet over de studierichting informatica. Een afstudeerrichting b.i. zou derhalve aan de E.U.R. niet kunnen worden geïmplementeerd. In de regio Leiden, Delft en Rotterdam, waarin de drie in deze steden gevestigde instellingen van w.o. samenwerken op het gebied van de informatica, is men van mening dat de inbreng van de enige economische faculteit in die regio, nl. die aan de E.U.R., noodzakelijk is voor een efficiënte en effectieve opbouw van de b.i. in die regio. De discussie gaat op dit moment vooral over de vraag of een experimentele studierichting het geëigende middel is om de beoogde samenwerking te realiseren. Het voorstel van de E.U.R. is door de

A.R. in de vergadering van 25 maart j.l. overgenomen nadat overigens een zgn. toetsingscommissie negatief had geadviseerd.

Het moet in hoge mate worden betreurd dat de besluitvorming rond het onderwijs in de informatica en mogelijke toepassing daarvan binnen het wetenschappelijk onderwijs wordt gekenmerkt door besluiteloosheid die resulteert in een grote vertraging en het ontstaan van achterstanden t.o.v. andere landen. Zo dateert een eerste voorstel tot het instellen van een studierichting informatica uit 1972. In 1974 heeft de A.R.S.I. een nieuw verzoek met een compleet programma ingediend. In dat verzoek wordt gewezen op de noodzaak tot het instellen van een studierichting b.i.k. De studierichting informatica werd ingesteld in 1981. Een verzoek tot het instellen van een studierichting b.i.k. is door de sectie economie gedaan in juni 1981, dit verzoek is herhaald in 1982.

3 SPECIALISATIES EN TOEPASSINGEN

Een deel van de in de vorige paragraaf genoemde besluiteloosheid is een gevolg van het ontbreken van een duidelijke strategie bij het wetenschappelijk onderwijs. Naast het ontbreken van een strategie is er ook geen eenstemmigheid over het gebruik van termen en over de wijze waarop in het bijzonder de toepassingen van de informatica moeten worden geïmplementeerd. Als illustratie van de verwarring rond termen geven wij als voorbeeld drie definities van informatica.

- De A.R.S.I. omschrijft informatica als: de wetenschap van het verwerken van informatie m.b.v. automaten [2].
- In het zgn. V.I.N.-rapport wordt informatica omschreven als: de theoretische studie van informatiesystemen te zamen met hun praktische realisatie in menselijke taken en machines [3].
- Brussaard noemt informatica: de brugdiscipline tussen technische wetenschappen en de bestuurs- en bedrijfswetenschappen [4].

De verschillen in mening rond termen zijn verklaarbaar door verschillen in visies t.a.v. de volgende factoren.

- a. De rol en de functie van de informatica in het onderwijs.
 - b. De multidisciplinaire functie van de informatica.
- a. De informatica kan op het terrein van het onderwijs een rol spelen op vier verschillende niveaus.
 1. Onderwijs moet, evenals processen in andere organisaties, worden georganiseerd. De computer kan daarbij worden ingeschakeld. Daarbij kan men denken aan een automatisering van belangrijke delen van de administratie, alsmede aan een ondersteuning van beslissingsprocessen, zoals bijv. het samenstellen van lesroostertabellen.
 2. De computer kan worden gebruikt als didactisch hulpmiddel, bijv. voor de illustratie van de te behandelen stof of als leermeester van de leerling, voor zover het het doceren van routinematige activiteiten betreft. Het voordeel van dit laatste is dat een op de leerling afgestemd systeem kan worden geïmplementeerd.
 3. In aansluiting hierop is het te verwachten dat in vele gevallen de computer zal worden ge-

bruikt als verlengstuk van de menselijke probleemoplosser. Daarbij gaat het er in eerste instantie om gebruik te maken van de computer voor het specificeren van empirische modellen. Voorts is het zeker mogelijk de computer ook te gebruiken voor het formuleren van conceptuele modellen, zie volgende punt. Op dit niveau worden potentiële gebruikers van informatiesystemen opgeleid. Deze gebruikers zijn geen computer- of informatica-deskundigen, zij zijn probleemgericht bezig. De computer wordt gebruikt voor het opheffen van menselijke cognitieve restricties.

4. In een aantal vakgebieden zullen deskundigen – zo men wil specialisten – moeten worden opgeleid om het raakvlak tussen informatica en toepassing te specificeren en te implementeren. Deze specialisaties zijn noodzakelijk om de volgende redenen:

1. De kennis vereist voor het implementeren ligt zowel op het gebied van de toepassing als op dat van de informatica. De kans is groot dat die noodzakelijke kennis binnen de b.i. of in de bij de toepassing betrokken discipline niet wordt onderwezen. Op dit moment is dit bijv. het geval, naar onze mening, bij het onderwijs t.b.v. de programmatuur voor decision support systemen. Een specialisatie, zoals wordt beoogd met de b.i.k., tracht in deze lacune te voorzien.
2. Zoals in par. 2 reeds werd opgemerkt, zal de ontwikkeling van bepaalde disciplines mede afhangen van het gebruik dat bij de constructie van conceptuele modellen wordt gemaakt van de informatica. Daartoe is het vereist dat wordt beschikt over voldoende kennis van de informatica om het gewenste proces te kunnen starten en te kunnen continueren. Deze ontwikkeling zal vooral relevant zijn voor empirische wetenschappen. Ons standpunt komt in hoofdlijnen overeen met dat van de A.R.S.I.[5]. ARSI-330 neemt een meer gedetailleerd standpunt in t.o.v. diverse disciplines. Wij hebben ons beperkt tot enkele hoofdlijnen, waarbij het gebruik dat van de informatica kan worden gemaakt bij de uitbouw van bepaalde disciplines voor ons een belangrijke overweging is.

- b. Regelmatig komt men de gedachte tegen informatica te definiëren als een wetenschap die het mogelijk moet maken langs een multidisciplinaire weg problemen m.b.v. computers op te lossen. Op zich is de gedachte correct, alleen de weg die men inslaat is, naar onze mening, niet juist. Multidisciplinair krijgt geen inhoud door het bij elkaar voegen van vakken, zeker niet als men het methodologisch beziet. Multidisciplinair moet een inhoud krijgen door de ontwikkeling van methoden waarmee de relevante aspecten die men wil beschouwen ook *simultaan* kunnen worden verwerkt. De computer stelt ons in staat het arsenaal van methoden dat dit mogelijk moet maken te vergroten. Zekerheid daarvoor bestaat niet en de resultaten op dit terrein tot nu toe

zijn niet overweldigend. Duidelijk is wel, zo er al van resultaten sprake is, denk in dit verband bijv. aan simulatie, dat de inbreng van de informatica bij de constructie van nieuwe conceptuele modellen gering is geweest. De informatica te omschrijven als een multidisciplinaire wetenschap brengt geen oplossing. Men loopt zelfs het risico dat het mogelijke kind met het bekende badwater verdwijnt. Wij zijn van mening dat de voorgestelde specialisaties zich zullen moeten richten juist m.b.v. computers het multidisciplinaire van een inhoud te voorzien. In dat geval is er in elk geval enige kans dat die methoden ook in de betrokken disciplines zullen worden toegepast. Voorts hebben die specialisaties, zoals de ervaringen binnen het HBO leren, een uitstralingseffect op het gebruik van informatica bij de opleiding van potentiële gebruikers.

Voor het specificeren van specialisaties is het van wezenlijk belang dat een keuze wordt gemaakt over de rode draad die de vakken binnen die specialisatie moet verbinden[6]. Die keuze kan niet zijn gebaseerd, zoals Boersma suggereert, op een overeenkomst in termen of doeleinden [7].

Immers in een dergelijke situatie ontbreekt de rode draad. Niet het studieobject is relevant, maar de wijze waarop men het object wenst te bestuderen. Het is op dit punt dat de informatica, evenals in de economie de wetkunde, een grote rol kan spelen. Overigens kunnen er verschillende rode draden worden onderscheiden. Wij zullen enkele daarvan in de volgende paragraaf schetsen, waarbij onze voorkeur uitgaat naar een integrale benadering m.b.v. het begrip informatiesystemen.

4 ONTWIKKELEN VAN INFORMATIESYSTEMEN ALS PROBLEEM

Informatiesystemen worden ontwikkeld om bij te dragen tot de oplossing van problemen in organisaties.

Bij het ontwikkelen van informatiesystemen wordt vaak een parallel getrokken met de rol van een architect. Velen zullen het er over eens zijn dat het bedrijven van architectuur een combinatie is van kunde en kunst. Deze constatering gaat zeker op voor het ontwikkelen van informatiesystemen. De betiteling b.i.k. voor dit vakgebied zou de indruk kunnen wekken dat hier geen creatieve vaardigheden in het geding zijn. Het tegendeel is echter waar.

Het spanningsveld tussen kunst en kunde maakt het onderwijs in de b.i.k. niet eenvoudig. Het onderwijskundig probleem ligt niet zozeer in het bijbrengen van materie-kennis en vaardigheden om verschillende methodieken en methoden te kunnen hanteren. Natuurlijk verschillen diepgang en presentatievorm afhankelijk van de onderwijssituatie, bij het HBO of het wetenschappelijk onderwijs, of het regulier of cursorisch onderwijs. Het probleem ligt eerder in de samenhang die tussen de verschillende onderdelen van de b.i.k. moet worden aangebracht.

Een kader voor een mogelijke integratie willen wij aanreiken door het ontwerpen van informatiesystemen als een proces van probleemoplossen te beschouwen.

De complexiteit van dit probleem kan worden aangepakt door verschillende sub-problemen te onderscheiden: [8] – een systemisch probleem, dat zich met de mo-

dellering van een objectsysteem vanuit een organisatorische invalshoek bezighoudt:

- een infologisch probleem, waarbij de gegevensstructuren en de verwerkingsprocessen die nodig zijn om informatie te kunnen produceren centraal staan;
- een datalogisch probleem, dat zich richt op de wijze waarop en de vorm waarin het gegevensverwerkend systeem wordt gerealiseerd;
- een technologisch probleem, waarin men moet aangeven welke technische hulpmiddelen worden gebruikt.

Bubenko maakt een iets andere indeling door een 'understanding stage' en een 'logical data and programming model' te onderkennen [9]. Met Bubenko zijn wij van mening dat een integrale benadering van de problematiek van de constructie van informatiesystemen alle vier door ons onderscheiden sub-problemen moet omvatten. Juist het feit dat er van sub-problemen sprake is, maakt dat het buiten beschouwing laten van één of meer sub-problemen leidt tot een keuze van een rode draad met beperkte geldigheid.

Voor het schetsen van een samenhang tussen de sub-problemen maken we een onderscheid tussen methodieken, methoden en activiteiten. Een methodiek of modelcyclus definieert de volgorde waarmee bepaalde activiteiten voor het oplossen van een probleem moeten worden doorlopen.

Methoden definiëren verschillende mogelijkheden waarop activiteiten kunnen worden uitgevoerd. Wij zijn van mening dat de activiteiten noodzakelijk voor het oplossen van het sub-probleem systemlogie in principe verschillen van die noodzakelijk voor de oplossing van de andere sub-problemen. Een vergelijking van methodieken en methoden, die worden gehanteerd bij het ontwerp van informatiesystemen, leert dat het merendeel der methodieken en methoden zich richt op het infologisch, het datalogisch en het technologisch sub-probleem [10]. Het systemlogisch probleem krijgt nog relatief weinig aandacht. Daarnaast blijkt dat vele methodieken een volgorde van activiteiten aangeven, zonder zich uit te laten over de invulling ervan. Vele methodieken leggen het accent op het infologisch probleem of op de ontwikkeling van een 'logical data and programming model'. Wij willen de nadruk leggen op de 'understanding stage' of het systemlogisch probleem omdat:

1. het resulterende produkt deel gaat uitmaken van het object systeem en de effectiviteit hiervan uitermate belangrijk is;
2. de complexibiliteit van applicaties op het gebied van de transactieverwerking verder toeneemt – denk aan procesbesturing en operationele planning – en daarmee de noodzaak voor een goede organisatorische beschrijving dringender wordt;
3. alleen op deze manier de ontwikkeling van Decision Support Systemen (DSS) vanuit de transactieverwerkende processen gestalte kan krijgen;
4. het waarschijnlijk is dat een standaardisatie van methoden binnen de infologie alleen kan worden bereikt als samenhangen met het systemlogisch sub-probleem worden onderkend en gespecificeerd.

Bij de ontwikkeling van informatiesystemen zijn niet alleen methodieken en methoden belangrijk. Met name zal een oplossing van het systemlogische sub-probleem al-

tijd, impliciet of expliciet, zijn gebaseerd op een paradigma, worldview of ontwerpfilosofie betreffende het functioneren van organisaties. In het bijzonder vanuit die filosofie zullen de activiteiten beeldvorming, de constructie van een conceptueel model, en een probleemspecificatie, de constructie van een empirisch model, van een inhoud moeten worden voorzien.

De eerste stap voor de oplossing van een probleemsituatie is de activiteit beeldvorming, waarbij men verschillende filosofieën kan hanteren. Zonder naar volledigheid te willen streven noemen we enkele benaderingen om tot een conceptueel model te komen.

1. Bij het beschrijven van een objectsysteem kan men uitgaan van vooraf gedefinieerde functies. Vaak wordt hiervoor de traditionele indeling marketing, productie en financiering gehanteerd. De afstemming van de verschillende functies verloopt dan via het top-management en/of via een budgetteringssysteem. In deze gedachtengang ligt een opzet van een informatiesysteem dat gebaseerd is op bijv. de goederenstroombeheersing niet voor de hand.
2. Verrijn Stuart introduceert in navolging van Blumenthal een besturingsmodel van een organisatie. Ook Bemelmans slaat deze weg in. [11]. Dit kader werkt verhelderend voor de identificatie van een objectsysteem en het informatiesysteem. Echter, de inhoud van de componenten en de subsysteem-keuze wordt niet door een besturingsmodel aangedragen. Bovendien leidt dit denken in regelkringen mogelijk tot het verwaarlozen van een belangrijk aspect bij beslissingen in organisaties, namelijk het genereren van en kiezen uit alternatieven.
3. In de wereld van DSS wordt er veelvuldig van uitgegaan dat 'zwak gekoppelde' systemen een goede oplossing bieden voor afstemmingsproblemen in organisaties. Men denkt dan individuen of groepen die met personal computing hun beslissingen ondersteunen, te kunnen afstemmen via een budgetteringssysteem of via een data-communicatienetwerk met mogelijkheden voor elektronisch berichtenverkeer etc. Afstemming expliciet rekening houdend met een coördinatie uitgangspunt kan hier waarschijnlijk moeilijk gerealiseerd worden en wildgroei lijkt niet onlogisch, tenzij hiervoor in het systemlogisch ontwerp waarborgen worden ingebouwd [12].
4. Bij het ontwikkelen van informatiesystemen wordt vaak gesteld dat meer en tijdige informatie tot betere besluitvorming leidt. Wij concluderen dat de beschikbaarheid van gegevens voor beslissingen die op het operationele vlak worden genomen niet noodzakelijk tot meer inzicht leidt, met name wanneer bijvoorbeeld geaggregeerde cijfers gegeven worden zoals in de praktijk gebruikelijk is. Wij kunnen in deze gevallen toch een oplossing vinden door op simulatie gebaseerde beslissingsondersteunende systemen te hanteren, die mogelijkheden bieden om op detailniveau uitspraken te doen [8].

Deze lijst van punten is niet uitputtend bedoeld, wij trachten ermee aan te geven dat de ontwerpfilosofie op systemlogisch niveau uitspraken veronderstelt, die van wezenlijk belang zijn voor de specificatie van de daarop vol-

gende sub-problemen. Bij de vraag naar de keuze van de rode draad voor het verbinden van de onderwerpen die binnen de b.i.k. worden behandeld, zullen wij drie benaderingen onderscheiden.

1. Zoals reeds opgemerkt, wordt het merendeel van het onderwijs in de b.i.k., ook in het buitenland, afgestemd op een startpunt vanuit het infologisch sub-probleem. Centraal staat dan de wijze van beschrijving van de bestaande toestand, veelal op het gebied van de dataverwerking, in organisaties. Men suggereert dat met die methoden een analyse kan worden uitgevoerd die leidt tot een ontwerp. Wij betwijfelen dat, met name het punt van de analyse. Altijd blijft onduidelijk hoe die analyse, methodisch gezien, wordt uitgevoerd. Evenmin wordt duidelijk hoe vanuit de beschrijving het ontwerp tot stand komt. Het is wel mogelijk het ontwerp op eenzelfde manier te beschrijven en daarmee de bestaande, beschreven, toestand te vergelijken met de gewenste, ontworpen, toestand. Uit een dergelijke vergelijking kunnen de nodige conclusies worden getrokken, bovendien is een bepaalde methode van specificeren voor het maken van een ontwerp op zich een groot pluspunt. Een ander voordeel van deze benadering is dat de b.i.k. vrij op zichzelfstaand kan worden gespecificeerd.

Nodig zijn de vakken informatica, voor het aandragen van gereedenschappen voor het ontwerp, de technieken voor het maken van een beschrijving, veelal aangeduid met de benaming systeemanalyse en systeemontwerp en één of andere ontwerpfilosofie. De ontwerpfilosofie is vanuit een bepaald vakgebied afkomstig bijv. de administratieve organisatie of een theorie over informatiesystemen. In de hiervoor gesignaleerde vier opvattingen betreffende de constructie van een beeld van de organisatie komt duidelijk naar voren dat de gehanteerde filosofieën zich niet lenen voor een onderlinge vergelijking. Het grote bezwaar dat tegen deze keuze van de rode draad kan worden aangevoerd is dat een gemeenschappelijk uitgangspunt ontbreekt, zie ook punt 3.

2. De tweede benadering is die vanuit de wiskundige systeemtheorie en/of het operationeel onderzoek. Bij deze benadering staat de problematiek van de constructie van een model centraal. Veelal ligt de nadruk op het conceptuele model dat als een ontwerp wordt gehanteerd. Als voorbeeld kan het gebruik van het besturingsparadigma in relatie met de systeemtheorie worden genoemd. Empirische modellen worden vrijwel nooit als basis voor een ontwerp gebruikt, hooguit als een beschrijving van een bestaande toestand. In beide gevallen ontbreken duidelijke verbanden met het infologische sub-probleem en de wijze waarop een ontwerp voor dat probleem zou moeten worden gemaakt.
3. De derde benadering, die wij in Groningen hanteren, berust op een aanpak vanuit de systemologie, waarbij conceptuele modellen berustend op het paradigma van beperkt rationeel handelen van organisaties worden gehanteerd. Daarbij, zie volgende paragraaf, wordt expliciet een verband gelegd met de problematiek van het beschrijven van bestaande toestanden met empirische modellen en het maken van een ontwerp voor het oplossen van het probleem en het informatiesysteem dat daarvoor nodig

is. Het voor- en nadeel van deze benadering is dat de generalisatiegraad van het paradigma van beperkt rationeel handelen laag is. Het voordeel daarvan is dat men wordt gedwongen *een* invulling te geven, het nadeel is dat anderen die invulling doorgaans niet beschouwen als *de* invulling en het dan om bepaalde redenen niet wensen te accepteren. Wij zijn van mening dat het paradigma van beperkt rationeel handelen een veelheid van conceptuele en empirische modellen niet in de weg staat, maar daartoe juist uitnodigt. Wel moet de combinatie conceptueel en empirisch worden ingevuld. Die combinatie maakt echter een discussie over de inhoud mogelijk en het creëert de mogelijkheid een verband te leggen met het maken van een ontwerp voor het systemologische en infologische sub-probleem. Het zijn vooral deze redenen die ons hebben doen kiezen voor deze derde benadering [13].

5 EEN ONDERWIJSKUNDIGE INVULLING

Het creatief hanteren van methodieken, methoden en ontwerpfilosofieën speelt een belangrijke rol bij het ontwikkelen van informatiesystemen. De b.i.k. heeft te maken met:

- de keuze van een paradigma en een ontwerpfilosofie, die vertaald moeten worden in een aantal bouwstenen en gereedenschappen;
- de keuze van een methodiek in een specifieke probleemsituatie, die aangeeft in welke volgorde de activiteiten in het ontwerpproces worden doorlopen en welke beschrijvingswijzen mogelijk zijn om de huidige en de gewenste situatie weer te geven;
- de keuze uit methoden voor de analyse van de bestaande situatie en het genereren van nieuwe alternatieven, waarbij de organisatorische mogelijkheden in het oog gehouden moeten worden.

In de reeds 8 jaren formeel bestaande doctorale afstudeerrichting Bestuurlijke Informatica aan de Economische Faculteit te Groningen wordt de derde in de vorige paragraaf genoemde benadering gevolgd. De ervaringen met deze afstudeerspecialisatie kunnen als volgt worden vertaald in een studierichting b.i.k. (zie tabel 1).

Bij deze vertaling moet het volgende worden opgemerkt.

- a. De vertaling past in de algemene structuur die door de sectie economie, zie de laatste kolom van tabel 1, is gegeven. Die structuur laat duidelijk de mogelijkheid open en daarvoor is bewust gekozen, een zekere mate van vrijheid bij de invulling te geven. De invulling in tabel 1 gaat uit van het bestaan van een studierichting b.i.k.
- b. Binnen de studierichting b.i.k. zijn geen afstudeerrichtingen onderscheiden. Verwacht mag worden dat die t.z.t. wel kunnen ontstaan.
- c. De tweede fase, in het bijzonder de onderzoekopleiding, is niet gespecificeerd. De sectie economie en A.R.S.I. stellen zich voor, voor zover de momenteel geldende wet wordt uitgevoerd, hiervoor gezamenlijk een programma te construeren [14].

In tabel 2 is een nadere invulling gegeven van tabel 1. Daarbij kan het volgende nog worden opgemerkt. Uitgangspunt is de in de vorige paragraaf genoemde derde benadering waarbij centraal staat de systemologische pro-

Tabel 1: Studierichting b.i.k. uren tabel

Vak	Propadeuse		Doctoraal				Totaal		Voorstel sectie econo- mie	
	weken	%	2e jr	3e jr	4e jr	Totaal		weken	%	% van het doctoraal progr.
			weken	weken	weken	weken	weken			
1. Bedrijfseconomie	9	21	16	16	-	32	25	41	24	25
2. Algemene economie	9	21	12	-	-	12	10	21	13	10
3. Informatica	8	19	8	10	12	30	24	38	23	22.5
4. Wiskunde	10	24	-	4	6	10	8	20	12	7.5
5. Statistiek	4	10	6	4	-	10	8	14	8	7.5
6. Administratieve organisa- tie	-	-	-	6	6	12	10	12	7	10
7. Boekhouden	2	5	-	-	-	-	-	2	1	-
8. Vrije keuze	-	-	-	2	18	20	15	20	12	17.5
	42	100	42	42	42	126	100	168	100	100

Tabel 2: Studierichting b.i.k. inhoud vakken

Vak	Jaar	Omvang	Inhoud
1. Bedrijfseconomie	1	9	Introductie, Financiering, Marketing, Organisatie/Leiding, Bestuurlijke Informatiekunde
	2	16	Financiering, Marketing, Organisatie/Leiding, Bestuurlijke Informatiekunde
	3	6	Systemen
	10	10	Beslissingsondersteunende systemen
2. Algemene economie	1	9	Samuelson, Openbare Financiën
	2	12	Prijstheorie, Macro-economie, Intern. Economische Betrekkingen
3. Informatica	1	8	Programmeren Pascal, terminalvaardigheid, Applicatieprogramma- tuur, Bestandsorganisatie, Micro-computer, Hanteren applicatiepro- grammatuur
	2	8	Programmeren, Bestandsorganisatie, Datacommunicatie, Bedrijfssys- temen, Architectuur
	3	10	Verdieping, onderwerp naar keuze
	4	12	Afstudeeropdracht
4. Wiskunde	1	10	Analyse, Lineaire Algebra, Logica, Discrete Wiskunde
	3	4	Operationele Research
	4	6	Informatieanalyse
5. Statistiek	1	4	Algemene en Economische Statistiek
	2	6	Toetsen en Methodologie
	3	4	Simulatie
6. Administratieve organisatie	3	6	Administratieve Organisatie
	4	6	S.A.S.O.
7. Boekhouden	1	2	Inleiding
8. Vrije keuze	3	2	Keuze afhankelijk van specialisatie:
	4	18	b.v.: bedrijfseconomie boekhouden administratieve organisatie.

bleembeschrijving van beslissingsprocessen in organisaties en de vraag hoe hier met computers tot een betere besluitvorming kan worden gekomen. De beschrijving van beslissingsprocessen in verschillende modelvormen en de voeding van deze modellen met al dan niet in de organisatie aanwezige gegevens staan centraal. Dit ge-

beurt reeds in de bedrijfseconomische vakken die in het eerste en tweede jaar worden gegeven. In het derde jaar wordt dit uitgewerkt in een semestercursus Systemen en Simulatie, die gevolgd wordt door een semestercursus Beslissingsondersteunende Systemen. Parallel hieraan wordt de nodige diepgang in de Informatica opgedaan.

Daarnaast staan de vakken Wiskunde, Statistiek en Administratieve Organisatie.

De materie- en modelkennis wordt samengebracht in de semestercursus Systeemanalyse en systeemontwerp (S.A.S.O.). Hier worden aan de hand van casusposities verschillende methodieken, methoden en ontwerpfilosofieën geanalyseerd. Vaardigheid wordt bijgebracht in het hanteren van enkele hiervan. In de opleiding wordt veel nadruk gelegd op de evolutionaire ontwikkeling van beslissingsondersteunende systemen en informatiesystemen en op prototyping (zie de cursus beslissingsondersteunende systemen). Centraal staat, en dit fungeert als rode draad, het uitgangspunt dat een bestaande situatie eerst moet worden beschreven en gevalideerd, voordat het ontwerpproces start. Daarmee wordt een referentiekader geïntroduceerd, waarmee een participatieve aanpak kan worden gerealiseerd. Keuzevrijheid bestaat er met betrekking tot een bedrijfs-economisch vak, de vrije keuze, in de invulling van het derde en vierde jaars vak Informatica en in de afstudeeropdracht. Mogelijke specialisaties zijn bijvoorbeeld ontwerptechnieken, DSS, organisatieverandering, EDP audit.

6 DE SITUATIE IN HET BUITENLAND

We beperken de bespreking betreffende het buitenland tot twee landen, nl. de V.S. en de bondsrepubliek Duitsland en tot twee onderwerpen, nl. in hoeverre komt b.i.k. als een zelfstandig programma voor en wat is de rode draad die men bij het formuleren van het programma heeft gekozen [15]. We beginnen met het laatste onderwerp.

Zowel in de V.S. als in Duitsland blijken met b.i.k. vergelijkbare programma's voor te komen. Het verschil met Nederland is het grote aantal programma's waarin informatica en/of informatiesystemen als verplicht vak in een relatief grote omvang is opgenomen. Het is daarom moeilijk een duidelijke grens te trekken tussen b.i.k. en die andere programma's zonder de problematiek van de rode draad in de beschouwing te betrekken. De verschillen die bestaan, zie tabel 3 en 4, zijn voor een deel verklaarbaar door verschillen in onderwijssystemen tussen de V.S., Duitsland en Nederland. Het systeem in de V.S. is van die drie het meest flexibel en kent een grote verscheidenheid in programma's met eenzelfde naam. Tabel 3 geeft een overzicht van aantallen programma's in de V.S. en die, welke voldoen aan de minimum eisen die zijn ontleend aan het ACM Information System Program [16].

In de Duitse publicatie worden drie alternatieve programma's onderscheiden, die verschillen naar het aan-

deel dat informatica c.q. informatiesystemen deel uitmaken van het programma [17]. Alternatief 2 en alternatief 3 zijn op een groot aantal punten vergelijkbaar met het door de sectie economie voorgestelde programma, voor zover het de omvang van de informaticacomponent betreft. In alternatief 1 wordt aan informatica weinig of geen aandacht besteed als verplicht vak, als keuzevak heeft het een omvang van minder dan 10 semesteruren. Op bepaalde onderdelen van die opleiding, bijv. bij het onderwerp econometrie en corporate models, is het niet uitgesloten dat vrij veel gebruik wordt gemaakt van computers. Alternatief 2 dekt het traject van 10-20 verplichte semesteruren in de informatica en/of informatiesystemen, alternatief 3 omvat meer dan 20 semesteruren. Tabel 4 is gebaseerd op een enquête bij de 54 in Duitsland, Oostenrijk en Zwitserland in aanmerking komende instellingen van wetenschappelijk onderwijs. Van de 54 hebben er 49 geantwoord, de percentages in tabel 4 hebben daarop betrekking. De cijfers verwijzen naar de situatie in het jaar 1980.

Tabel 4

	perc.
alternatief 1	30
alternatief 2	60
alternatief 3	10

Vergeleken met Nederland kan worden gesteld dat alternatief 2 binnen het wetenschappelijke onderwijs in veel mindere mate voorkomt en dat het merendeel van de 'opleidingen' in Nederland kan worden geplaatst in alternatief 1. De situatie in de V.S. schijnt vergelijkbaar te zijn met die in Duitsland, alhoewel tussen de onderwijssystemen van beide landen vrij grote verschillen bestaan. Men gelieve dit mede in de beschouwing van de volgende cijfers te betrekken. De enquête in de V.S. berust op het aanschrijven van:

205 business schools – antwoordpercentage 53;

149 computer science departementen – antwoordpercentage 49;

159 collegiate chapters of the ACM – antwoordpercentage 11.

Uit het onderzoek van het materiaal bleek dat 91 instellingen de in tabel 3 genoemde opleidingen aanboden.

Tabel 3 en tabel 4 laten duidelijk zien dat de stelling dat het niet ontwikkelen van integratiemogelijkheden met informatica leidt tot een blokkering van de toepassingen voor Nederland wordt bevestigd. Het verschil met Duitsland is daarbij bijzonder illustratief.

De door ons geschetste problematiek van de rode draad wordt in beide andere landen onderkend en leidt daar tot dezelfde problemen als door ons geschetst. De volgende twee citaten kunnen onze conclusie illustreren.

Tabel 3

Aard van het programma	Aantal progr. met als onderwerp informatiesystemen	Programma's die voldoen aan de minimum eisen	
		bij Business of Management college	bij Computer Science of Engineering college
Bachelors	70	42	11
Masters	54	25	9
Doctoral	28	niet geëvalueerd	

Nunamaker stelt: 'Another aspect of the project was to identify problem areas in the curriculum. The single most critical problem area appeared to be contents of the system analysis and design courses. Many new techniques such as structured design and top-down approaches are available, but systems analysis and design instruction appears still to be on an *ad hoc* basis. Course description and content revealed very little commonality in what is taught in systems analysis and design courses. Content ranged from the very general overview (a sort of 'gee whiz' course) to a 'nitty-gritty, nuts and bolts' type of course, largely ignoring the problems of analysis and design' [18].

Seibt constateert: 'Die Gebiete Betriebs- und Wirtschaftsinformatik sind einfach noch zu jung, als das eine einheitliche oder wenigstens von vielen akzeptierte Begriffsabgrenzung erwartet werden kann. Von der Mehrzahl der Fachvertreter wird heute die Meinung vertreten, das diese Gebiete eher der Betriebswirtschaftslehre bzw. den Wirtschaftswissenschaften als der Informatik zuzurechnen sind' [19].

Het eerste deel van de uitspraak van Seibt onderschrijven wij niet. Het te jong zijn vinden wij geen argument voor het hanteren van onduidelijkheid over de inhoud van het curriculum.

7 SAMENVATTING

Wij zouden het voorgaande in de volgende punten willen samenvatten.

1. B.i.k. behoort een studierichting te zijn om te bereiken dat:
 - a. specialisten kunnen worden opgeleid voor de verdere integratie van informatica in toepassingsgebieden;
 - b. specialisten kunnen worden opgeleid voor het ontwerp en de implementatie van informatiesystemen, waarbij zal een zekere overlap kunnen bestaan met de b.i. opleiding in de studierichting informatica. Een dergelijke overlap achten wij gewenst;
 - c. de kennis binnen instellingen van wetenschappelijk onderwijs kan worden verkregen voor de opleiding van potentiële gebruikers, het tweede alternatief in tabel 4. Het moet sterk worden betwijfeld of in de situatie in Nederland het tweede Duitse alternatief langs een andere weg realiseerbaar is. De ontwikkeling geeft weinig argumenten dat wel te veronderstellen.
2. Er moet meer aandacht worden besteed aan het specificeren van een samenhang binnen curricula betreffende b.i.k. Wij zijn geen voorstander van één samenhang, maar wel van een samenhang.

8 NOTEN

- [1] Studierichtingen en faculteiten, c.q. afdelingen van een T.H., vallen niet samen. Een faculteit (afdeling) kent meestal meerdere studierichtingen.
- [2] *Een curriculum voor de opleiding tot informaticus*, ARSI-280, Den Haag, Academische Raad, 1980.
- [3] *Over informatica-onderwijs, een verkenning*, Den Haag, Staatsuitgeverij, 1981.
- [4] Brussaard, B. K.; 'Elektronica en informatica. Kritische beschouwing van het Rapport van de Commissie Rathenau', *Informatie*, jrg. 22 (1980), nr. 1.
- [5] *Toepassingsgerichte Informatica en Informatica als Bijvak*, ARSI-330, Den Haag, Academische Raad, 1982.
- [6] Zie voor een uitvoerige beschouwing over de noodzaak daartoe, Bosman, A.; *Bedrijfseconomie en bestuurlijke informatica*, Leiden, Stenfert Kroese, 1978.
- [7] Boersma, S. K. Th.; 'Bestuurlijke informatiekunde en bedrijfskunde', *Informatie*, jrg. 3 (1983), nr. 25.
- [8] Sol, H. G.; *Simulation in Information Systems Developments*, Dissertatie, Rijksuniversiteit, Groningen (1982).
- [9] Bubenko, J. A.; CIAM, In: Olle, T. W., Sol, H. G., Verrijn-Stuart, A. A. (eds.), *Information Systems Design Methodologies: A Comparative Review*, North-Holland, 1982.
- [10] Zie T. W. Olle, H. G. Sol and A. A. Verrijn Stuart: t.a.p.
- [11] Bemelmans, T. M. A.; *Bestuurlijke informatiesystemen en automatisering*, Leiden, Stenfert Kroese, 1981.
- [12] Bosman, A.; 'Organizations, co-ordination and DSS', In: *Processes and tools for decision support*, ed. H. G. Sol, Amsterdam, North-Holland, 1983.
- [13] Het valt buiten het kader van dit artikel over de inhoud van de benadering meer in detail te gaan. Belangstellende lezers worden verwezen naar: A. Bosman: Systemen en methodologie I, II en III, *Informatie* jrg. 16 (1974), nr. 7/8, jrg. 19 (1977) nr. 1, jrg. 19 (1977) nr. 10. Een uitvoerige beschouwing over de methodische consequenties van het paradigma van beperkt rationeel handelen wordt gegeven in A. Bosman: *Een metatheorie over het gedrag van organisaties*, Leiden, Stenfert Kroese, 1977. In het proefschrift van Sol, zie noot 8, wordt uitvoerig ingegaan op de relaties met de ontwerp problematiek.
- [14] Zie A.R.S.I.-330, voetnoot 5.
- [15] Onze bespreking van beide landen is gebaseerd op voor de V.S. J. F. Nunamaker: 'Educational programs in information systems', *Communications of the ACM*, vol. 24 (1981), nr. 3. Voor de Bondsrepubliek Duitsland op *Betriebs- und Wirtschaftsinformatik*, eds. J. Griese e.a., Braunschweig, Fr. Vieweg und Sohn, 1981.
- [16] Ontleent aan Nunamaker, t.a.p., p. 132.
- [17] Seibt, D.; 'Stand der Betriebs- und Wirtschaftsinformatik Ausbildung an deutschsprachigen wissenschaftlichen Hochschulen', *Betriebs- und Wirtschaftsinformatik*, t.a.p. We hebben de alternatievenspecificatie van Seibt aangepast.
- [18] Nunamaker, J. F., t.a.p., p. 132/133.
- [19] Seibt, D., t.a.p., p. 25.

KENNIS EN ERVARING ROND HET ONTWERPEN VAN INFORMATIESYSTEMEN*

door prof. dr. H. G. Sol

Alhoewel computers een steeds groter deel van ons leven gaan uitmaken, stuit het gebruik hiervan in organisaties zowel op verbazing als op verwazing. Veel mensen op verschillende niveaus in de organisatie verbazen zich over de ongekende toepassingsmogelijkheden van informatietechnologie in onze informatiemaatschappij. De vraag naar het aangeven van richtingen in deze onduidelijke, onzekere en snel veranderende situatie wordt meer en meer gehoord. Voor anderen is dit geen vraag. Zij hebben een ongebreideld geloof in technologische mogelijkheden en vooruitgang en zien – soms verwaasd – geen belemmeringen op deze weg. Zijn beide houdingen echter gegrond? Of anders geformuleerd: Welke expertise hebben wij eigenlijk over het gebruik van computers voor het oplossen van kleine en grote problemen die mensen ontmoeten in organisaties? Deze vraag wordt in deze bijdrage toegespiest op het ontwerpen van informatiesystemen.

1 INLEIDING

Het lijkt mij goed allereerst enige begrippen te bespreken en een nadere begrenzing aan te brengen. Vervolgens wil ik aan de hand van een aantal paradoxen nagaan welke expertise ons ter beschikking staat. Dit geeft mij de basis voor het ontvouwen van enkele gedachten over nieuwe aanzetten voor het ontwerpen van informatiesystemen. Tenslotte wil ik ingaan op de betekenis hiervan voor een onderzoek- en onderwijsprogramma op dit jonge vakgebied.

2 BEGRIPPEN EN BEGRENZING

Ontwikkelingen rond informatiesystemen voltrekken zich in een hoog tempo. Generaties van computersystemen, van formele talen en van gereedschap volgen elkaar snel op. Ik wil hier niet het ontwikkelingspad van 1e via 4e en 5e naar n-de generaties van systemen schetsen. Ik stel voor het gebruik van het woord generatie in dit verband af te schaffen. De technologische ontwikkelingen gaan daarvoor te snel. De genoemde 4-5 generaties volgen elkaar op binnen 1-2 generaties gebruikers. Het lijkt mij aan te bevelen het woord generatie alleen in deze laatste context te hanteren. Immers, een gebruikersgeneratie heeft een kenmerkende achtergrond, opleiding, ervaring en belevingswereld. Ik vraag me af, of we niet te hoge verwachtingen hebben van de aanpassingsmogelijkheden binnen één gebruikersgeneratie in het licht van de technologische veranderingssnelheid.

In de ontwikkeling van het gebruik van computers in organisaties kan men een drietal fasen onderscheiden:

1. Geautomatiseerde gegevensverwerking (GGV);
2. Management informatiesystemen (MIS);
3. Beslissingsondersteunende systemen (BOS).

Met nadruk stel ik dat de namen GGV, MIS en BOS gebruikt worden ter aanduiding van verschillende stadia op

dit ontwikkelingspad, niet ter aanduiding van een onderverdeling in informatiesystemen.

2.1 Geautomatiseerde gegevensverwerking

Deze fase wordt gekenmerkt door de automatisering van massa-administraties. De ontwikkelingen worden beheerst door technische randvoorwaarden, met name beperkte rekensnelheid en geen direct toegankelijke geheugens voor de opslag van gegevens. De verwerking geschiedt seriegewijs met sequentieel toegankelijke geheugens.

De uitvoer is niet wezenlijk anders dan vóór de automatisering het geval is. De tijd noodzakelijk voor de productie van data kan worden ingekort. Daarmee kan in een aantal gevallen de frequentie van de informatievoorziening worden opgevoerd. Bij het ontwerpen van applicaties liggen de informatiebehoeften veelal vast. Centraal staat het volledig specificeren van de (administratieve) processen en de constructie van efficiënte programma's. De ondersteuning op het bestuurlijke vlak is gering, in het bijzonder bij de beslissingsvoorbereiding.

2.2 Management informatiesystemen

In de tweede fase wordt de computer meer en meer gebruikt voor planningdoeleinden. Dit wordt mogelijk door de toename van verwerkingscapaciteit en door direct toegankelijke geheugens. Van verschillende kanten, bijvoorbeeld uit de operationele research, worden algoritmen aangedragen waarmee planningproblemen op computers kunnen worden opgelost.

In het midden van de jaren zestig ontstaat het begrip management informatiesystemen. Door het merendeel van de auteurs wordt een MIS gedefinieerd, zie Burch en Strater [1974, p. 71], als een concreet systeem dat de volgende drie functies vervult:

- 'to meet legal and transactional data processing requirements;
- to provide information to management for support of planning, controlling, and decision making activities, and
- to provide a variety of reports, as required, to external constituents'.

* Deze bijdrage is een bewerking van de rede, getiteld 'Expertise rond Informatiesysteemontwerp', waarmee de auteur op 16 november 1984 het ambt van gewoon hoogleeraar in het ontwerpen van informatiesystemen aan de Technische Hogeschool Delft heeft aanvaard. De introductie is uitgegeven door Samsom Uitgeverij, Alphen aan den Rijn, 1984, onder ISBN 90 140 3444 X.

Kenmerkend voor deze fase is de veelal impliciet gehanteerde veronderstelling dat het functioneren van een organisatie kan worden beschreven door middel van gekoppelde modellen met verschillende niveaus van aggregatie die een totaal systeem vormen. Deze modellen bestaan meestal uit een stelsel vergelijkingen. De bruikbaarheid van deze modellen staat en valt met de mate waarin acceptabele stuurinformatie kan worden vastgelegd.

2.3 Beslissingsondersteunende systemen

De derde fase begint met de ontwikkeling van database systemen en interactieve programmatuur. Door de mogelijkheden van gegevensbanken, mens-machine interactie, data-communicatie en tekstverwerking kan direct worden aangesloten bij individuele (of groepsgerichte) beslissings- en gegevensverwerkende processen. Men poogt niet langer te verzanden in het top-down achterhalen van informatiebehoeften van managers (de GGV-aanpak) of in het beschrijven van beslissingsprocessen in verbanden (de MIS-aanpak).

Echter, BOS helpen managers bij het nemen van beslissingen, met name in niet duidelijk definieerbare probleemsituaties en ondersteunen, in plaats van vervangen het managementoordeel.

In Sol [1982] toon ik aan dat het ontwerpen van informatiesystemen, en ook van BOS, opgevat kan worden als een bijzonder geval van een proces van probleem oplossen. Immers, informatiesystemen (IS) bevatten én vervaardigen modellen van reële systemen (RS). Zelf vormen zij ook reële systemen.

Probleemoplossen is modelleren. Alhoewel er vele soorten modellen kunnen worden onderscheiden, maak ik in dit verband onderscheid tussen:

1. conceptuele en empirische modellen, en tussen
2. beschrijvende en voorschrijvende modellen.

Een beschrijvend empirisch model van het objectstelsel, zowel RS als IS, noem ik een KENmodel. Een ontwerp of doelsysteem is een empirisch model dat ik aanduid als MAAKmodel.

Als mensen een probleem onderkennen, vormen zij zich meestal op basis van een aantal veronderstellingen een beeld in de vorm van een plaatje of conceptueel model. Vaak gebeurt dit impliciet, maar altijd komt deze beeldvorming tot stand vanuit een bepaalde optiek, ook wel Weltanschauung genaamd. Een voorbeeld is: beschrijf de organisatie in termen van stromen van goederen, geld en mankracht.

Vervolgens wordt dan nagegaan of het gevormde beeld van de bestaande situatie empirisch hard gemaakt kan worden. Conceptuele modellen leggen datavrije visies vast. Empirische modellen geven een datagebonden beschrijving van een probleemsituatie. Van belang is, dat dit empirisch model een goede overeenkomst vertoont met de bestaande situatie. Het moet herkenbaar en valide zijn.

Door met dit kenmodel te manipuleren komt vaak het echte probleem boven tafel. In deze analyse-fase komen knelpunten naar voren. Echter, voor het zoeken naar een oplossing, in de synthese-fase, moeten suggesties worden aangedragen. Deze vloeien ten dele voort uit de analyse-fase via experimentatie. Een ontwerp-aanpak of organisatiefilosofie of speciale theorie, moet echter verdere aangrijpingspunten bieden. Nu gaat men voorschrijvende conceptuele en empirische modellen hante-

ren om tot een beter ontwerp in de vorm van een maakmodel te komen.

Het doorlopen van activiteiten in een proces van probleem oplossen, de modelcyclus, gebeurt vanuit een bepaalde optiek. Optiek en modelcyclus krijgen voor een bepaald toepassingsgebied concreet gestalte in een methodiek. Een methodiek geeft in een specifieke probleemsituatie aan in welke volgorde de activiteiten worden doorlopen en welke methoden en technieken hierin gehanteerd worden. Methoden zien wij als de werkwijzen die men in de activiteiten kan volgen met gebruikmaking van technieken als hulpmiddel.

Laat ik eens nagaan wat deze beschouwing nu betekent voor het ontwerpen van informatiesystemen, wanneer men dit door de bril van de informatiekundige bekijkt. Zoals gesteld is ontwerpen een proces van probleem oplossen. Omdat dit proces een onduidelijk gestructureerd en complex probleem is, wordt het vaak opgesplitst in hapklare brokken met een aanwijsbaar produkt. De produktieve aanwending van hulpmiddelen tijdens de stappen in dit proces is, naast kwaliteit, een belangrijk kenmerk van informatiesysteem ontwerp.

Het complexe ontwerp-probleem kan men aanpakken door:

1. Het probleem in sub-problemen te verdelen. Men gaat dan uit van een specifieke denkwijze en afbeeldingswijze.

Een voorbeeld is de indeling in:

- een systemologisch probleem, dat zich met de modellering van een objectstelsel vanuit een organisatorische invalshoek bezighoudt (het waartoe);
- een infologisch probleem, waarbij de gegevensstructuren en de verwerkingsprocessen die nodig zijn om de data te kunnen produceren centraal staan (het wat);
- een datalogisch probleem, dat zich richt op de wijze waarop en de vorm waarin het gegevensverwerkend systeem wordt gerealiseerd (het hoe);
- een technologisch probleem, waarin men moet aangeven welke technische hulpmiddelen worden gebruikt (het waar).

2. Een fasering in het ontwerp-proces aan te brengen. Men let nu op de werkwijze.

Davis e.a. [1980] stellen bij het ontwerpen informatiebehoeften centraal. Na een accurate beschrijving van informatiebehoeften zal het informatiesysteem met een grotere kans op succes, ontworpen en geïmplementeerd kunnen worden. Bij het bepalen van informatiebehoeften bestaan twee belangrijke bronnen van onzekerheid. De eerste ligt in het verkrijgen van een beschrijving van informatiebehoeften, bijv. d.m.v. interviews, analyse van beslissingsprocessen, etc. De tweede ligt in de verzekering dat de verkregen informatiebehoeften accuraat, consistent en volledig zijn. Factoren die invloed hebben op de mate van onzekerheid zijn bijvoorbeeld projectgrootte, mate van gestructureerdheid, begrip van de gebruikers en bekwaamheid van de ontwerpers. In ieder methodiek komt impliciet een strategie naar voren om onzekerheid te reduceren. Men kan verschillende strategieën onderscheiden om onzekerheid te reduceren, zie ook Sol en IJpelaar [1983]:

- directe acceptatie van gebruikersbehoeften;
 - een lineaire strategie;
 - een iteratieve strategie;
 - een incrementele of prototyping strategie.
3. Het project te besturen met de daarvoor beschikbare methoden en technieken. De invalshoek is nu de beheerswijze.

Probleem opsplitsing, fasering en projectbesturing vormen samen een projectaanpak, toegespitst op een specifieke probleemsituatie. Voor de duidelijkheid zij gesteld dat deze 3 aspecten zeker niet onafhankelijk van elkaar zijn. Uit een sub-probleemkeuze volgt een fasering en een controle-aanpak. In praktijksituaties van informatiesysteemontwerp kan hiervoor een aanzet worden gegeven door aan de hand van tabel 1 een zogenaamde 'contingency-analysis' uit te voeren.

	post onvoorzien
	laag _____ hoog
probleem	
structuur	
omvang	
gebruikers	
ontwerpers	

Tabel 1

Men schat de 'post onvoorzien' voor ieder van de factoren in. Bijvoorbeeld, een duidelijk probleem (laag), in een afgebakende structuur (laag), van grote omvang (hoog), met onervaren gebruikers (laag) en een redelijke ontwerpstaff (laag), vraagt om een lineaire strategie met accent op de infologie en de datalogie, terwijl de projectbeheersing en gebruikersopleiding uitermate belangrijk zijn. Is het probleem onduidelijk (hoog), in een niet vastliggende structuur (hoog), doch van beperkte omvang (laag), dan staat het systemologisch probleem centraal, waarvoor een incrementele strategie gehanteerd kan worden, indien de gebruikers zich hierin kunnen vinden (matig) met gekwalificeerde ontwerpers (laag). De fasen in de ontwikkeling van informatiesystemen zijn samenvattend in tabel 2 weergegeven.

Tabel 2

FASE	GGV	MIS	BOS
SOORT PROBLEMEN	(goed) gestructureerde processen	(goed) gedefinieerde informatie verzorging	slecht gestructureerde beslissingsondersteuning
OPTIEK of denkwijze	procesgericht, accent op datalogie en technologie	gegevensgericht, accent op infologie	objectgericht, accent op systemologie
MODEL CYCLUS of afbeeldingswijze	deductief	inductief	hypothetisch-inductief
METHODIEK of werkwijze	lineair	iteratief	incrementeel
PROJECTBESTURING of beheerswijze	projectgroep	harmonie	participatief

3 PARADOXEN ROND INFORMATIESYSTEMEN

Al vele jaren worden informatiesystemen ontworpen en geïmplementeerd. Diverse mensen zijn hierbij betrokken, zowel ontwerpers als gebruikers en gereedschapmakers. Het is opvallend dat over de ervaringen die bij deze groepen zouden moeten bestaan, weinig gepubliceerd wordt. Veel onderzoek richt zich op formele aspecten, bijvoorbeeld van ontwerpmethodieken. Empirisch onderzoek komt slechts langzaam van de grond, zie Bemelmans [1983]. Bovendien zijn uitkomsten hiervan vaak moeilijk generaliseerbaar. Toch wil ik een aantal ervaringsfeiten aan u voorleggen, en wel in de vorm van enkele paradoxen. Ik richt me achtereenvolgens op de denkwijze en afbeeldingswijze, op de werkwijze en op de beheerswijze bij het ontwerpen van informatiesystemen.

Denkwijze en afbeeldingswijze

Als men kijkt naar het onderscheiden van sub-problemen in een probleemsituatie stuit men op de paradox van KENNEN en MAKEN.

In de praktijk van de automatisering staat het snel kunnen maken van applicaties binnen beschikbare budgetten en mankracht centraal. Produktiviteit neemt een belangrijke plaats in tijdens het proces van ontwerpen, bouwen en implementeren.

Het oude adagium 'bezint eer ge begint', dat mede door de Japanse invloed op de informatisering wel als iets nieuws naar voren wordt geschoven, onderschrijft echter de aandacht voor het kennen van de uitgangssituatie om tot een goede kwaliteit van het nieuwe produkt te komen. Analytisch kijkend naar informatiesystemen heeft Langefors [1966] al vele jaren geleden, vanuit soortgelijke overwegingen, twee probleemgebieden onderscheiden, die hij aanduidt met de termen infologie en datalogie. Welke [1977] heeft hiernaast het systemologisch probleem onderscheiden. Hieraan is het technologische probleemgebied toegevoegd, zie Sol [1982].

In het licht van de technologische ontwikkelingen is het te verwachten dat de nadruk meer en meer naar het systemologische en infologische probleemgebied verschuift. Wellicht is het daarom aan te bevelen voortaan bij het ontwerpen alleen te spreken van een probleem om het objectstelsel te kennen en van een probleem om van hieruit een ontwerp en een implementatie te maken. Dit doet ook meer recht aan de onderkenning, dat het objectstelsel een dynamisch veranderend, open systeem is. Bovendien is er steeds minder sprake van het ontwerpen van geheel nieuwe informatiesystemen. Renovatie

van informatiesystemen zal belangrijker worden. Hier-voor is het kennen en herkennen van het objectssysteem essentieel. Maken gaat hier zeker niet zonder kennen. Voor het vinden van een juiste balans tussen kwaliteit en produktiviteit in het ontwerpproces wordt veelal teruggegrepen naar een methodiek. Methodieken kunnen zich in een grote belangstelling verheugen. Echter, vele methodieken besteden weinig aandacht aan het systemologisch probleem. Bubenko [1982] maakt als een van de weinigen onderscheid tussen een 'understanding stage' en een 'conceptual information modeling and conceptual database and program modeling stage'. De understanding stage speelt bij hem op het raakvlak tussen systemologie en infologie. Hier moet een volledig, beschrijvend empirisch model of kenmodel van het objectssysteem gemaakt worden. Een doorlichting van mogelijke knelpunten als basis voor het zoeken naar verbeteringen in een maakmodel, komt echter bij hem niet aan de orde. Ook de vraag welk conceptueel model meespeelt bij de empirische invulling wordt niet beantwoord. Hetzelfde kan opgemerkt worden van ISAC, zie Lundeberg [1982]. In ISAC wordt een stadium van change analysis onderscheiden. De overgang van beschrijvend naar voorschrijvend blijft ook hier in nevelen gehuld. Waar mogelijke knelpunten vandaan komen en hoe oplossingen hiervoor op hun merites moeten worden beoordeeld, blijft onduidelijk. Falkenberg et al. [1983] onderkennen dat ook in NIAM de beschrijving van het objectssysteem meer aandacht verdient. Zij stellen dan ook voor NIAM met ISAC te combineren. De genoemde bezwaren tegen ISAC met betrekking tot het voortraject worden hiermee echter niet weggelaten.

De methodiek BSP, zie Sebus [1981], legt zich toe op een strategische verkenning en de invulling van een informatieplan vanuit een top-down doelstellingsanalyse. De aanpak is deductief, voorschrijvend van aard en tamelijk gedetailleerd voor een vluchtige probleemherkenning en formulering. In D2S2, zie Van Lith [1982], wordt het stadium strategische analyse onderscheiden. Naast de onderkenning dat een informatieplan gestoeld moet zijn op een meer of minder gedetailleerde functie- en entiteitenanalyse, geeft deze methodiek niet aan hoe deze strategische analyse nu moet verlopen. Met andere woorden het systemologisch probleem wordt wel gesignaleerd, doch niet opgelost.

Naar mijn mening dient in het systemologisch probleem een herkenbare, verifieerbare en valideerbare beschrijving van het objectssysteem, zowel het RS als het IS, als bedrijfsmodel centraal te staan. De nadruk op de beschrijving van het objectssysteem en van het toepassingsgebied komen we ook tegen bij Bosman [1977], Budde en Sylla [1984], Kensing [1984], Nygaard et al. [1978]. Mumford [1984] wijst hier ook op. Zij geeft impliciet aan dat een voorwaarde voor een goed functioneren van haar 'participative approach' ligt in het scheppen van een herkenbaar, gemeenschappelijk referentiekader. Door een nauwkeurige beschrijving van het objectssysteem kan hieraan worden voldaan.

De nadruk op beschrijving brengt ons bij de paradox van INPUT en OUTPUT. Door anderen is opgemerkt dat output van een computer input is voor een gebruiker of besliser, en omgekeerd. Ik wil hier niet stilstaan bij de juistheid van deze naamgeving, maar wel bij de paradoxale vanzelfsprekendheid waarmee zowel beslissingsprocessen in organisaties als transformatieprocessen bin-

nen computers worden opgevat als een black-box.

Vele methodieken voor het ontwerpen van informatiesystemen leggen in een of andere vorm deze processen vast, waarbij meestal wordt teruggedeneerd van output naar input. Naast deze groep van 'proces-gerichte' methodieken kijken 'data-gerichte' methodieken naar informatiebehoeften en de gewenste datastructuren om aan deze behoeften te kunnen voldoen. Laat ik informatiebehoeften onderverdelen in behoeften gericht op instrumentele variabelen, op niet-beheersbare variabelen, op toestandsvariabelen en op uitgangsvariabelen. In beide groepen van methodieken wordt bij de vraag naar informatiebehoeften van gebruikers, de gewenste output, te veel aangenomen dat het voldoen aan deze behoeften ook tot betere beslissingen leidt. Dit gaat echter alleen op wanneer er empirisch een functioneel verband, ook wel effectiviteitsrelatie genoemd, tussen uitgangsvariabelen en andere variabelen bestaat. In vele methodieken wordt hier impliciet van uitgegaan. Echter, in Sol [1982] heb ik erop gewezen dat deze veronderstelling vaker *niet* dan *wel* opgaat. Kortom, beide scholen van aanpak staan en vallen met de beschikbaarheid en validiteit van functionele verbanden. Een uitweg wordt geboden in methodieken die uitgaan van een objectgerichte specificatie, zie Sol [1982]. Ik kom hier in de volgende paragraaf op terug.

Ik wil de nadruk leggen op de 'understanding stage' of het systemologisch probleem omdat:

1. het resulterende produkt als informatiesysteem deel gaat uitmaken van het objectssysteem en de effectiviteit hiervan uitermate belangrijk is;
2. de complexiteit van applicaties op het gebied van de geautomatiseerde gegevensverwerking verder toeneemt – denk aan procesbesturing en operationele planning – en daarmee de noodzaak voor een goede organisatorische beschrijving dringender wordt;
3. alleen op deze manier de ontwikkeling van BOS vanuit de gegevensverwerkende processen gestalte kan krijgen;
4. het waarschijnlijk is dat een standaardisatie van methoden binnen de infologie alleen dan kan worden bereikt als samenhangen met het systemologisch probleem worden onderkend en gespecificeerd. Met name geldt dit voor het theoretisch nog weinig onderbouwde begrip modularisatie.

Bij de ontwikkeling van informatiesystemen zijn niet alleen methodieken en methoden belangrijk. Met name zal een oplossing van het systemologische probleem altijd, impliciet of expliciet, zijn gebaseerd op een optiek of ontwerpfilosofie betreffende het functioneren van organisaties. In het bijzonder vanuit die filosofie zullen de activiteiten beeldvorming en probleemspecificatie van een inhoud moeten worden voorzien.

Voor de beeldvorming kan men verschillende filosofieën hanteren. Zonder naar volledigheid te willen streven noem ik enkele veel voorkomende benaderingen.

1. Bij het beschrijven van een objectssysteem kan men uitgaan van vooraf gedefinieerde functies. Vaak wordt hiervoor de traditionele indeling marketing, produktie en financiering gehanteerd. De afstemming van de verschillende functies verloopt dan via het top-management en/of via een budgetteringssysteem. In deze gedachtegang ligt een opzet van een informatiesysteem dat gebaseerd is op bijv.

de goederenstroombeheersing niet voor de hand. Immers, de functionele structuur van de organisatie staat hier niet ter discussie.

2. Verrijn Stuart [1976] introduceert in navolging van Blumenthal een besturingsmodel van een organisatie. Ook Bemelmans slaat deze weg in. Dit kader werkt verhelderend voor het onderscheiden van het RS en het IS. Echter, de inhoud van de componenten en de subsysteem-keuze wordt niet door een besturingsmodel aangedragen. Bovendien leidt dit denken in regelkringen mogelijkwijs tot het verwaarlozen van een belangrijk aspect bij beslissingen in organisaties, namelijk het genereren van en kiezen uit alternatieven. Anders gezegd, de nadruk ligt op een voorschrijvend conceptueel model, niet op een empirische invulling.
3. In de wereld van BOS wordt er veelvuldig van uitgegaan dat 'zwak gekoppelde' systemen een goede oplossing bieden voor afstemmingsproblemen in organisaties. Men denkt dan individuen of groepen die via persoonlijk computergebruik hun beslissingen ondersteunen, te kunnen afstemmen via een budgetteringssysteem of via een data-communicatienetwerk met mogelijkheden voor elektronisch berichtenverkeer etc. Afstemming expliciet rekening houdend met coördinatie kan hier waarschijnlijk moeilijk gerealiseerd worden en wildgroei lijkt niet onlogisch, tenzij hiervoor in het systemologisch ontwerp waarborgen worden ingebouwd.

Het is opvallend dat in deze benaderingen meer aandacht wordt gegeven aan een conceptuele beschrijving van het maakmodel, dan aan het genereren van alternatieve ontwerpen of doelsystemen. Over ontwerprichtlijnen wordt ofwel losjes heengelopen, ofwel worden uitgangspunten tot een geloof verheven. Het denken in 'zwak-gekoppelde systemen', is hiervan een voorbeeld.

Dat ontwerprichtlijnen nog een braakliggend terrein zijn voor empirisch onderzoek, kan geïllustreerd worden aan de paradox van AGGREGATIE. Bij het ontwerpen en invoeren van beslissingsondersteunende systemen in middelgrote en grote organisaties wordt meer en meer een logisch onderscheid gemaakt tussen

- een database, waarin gedetailleerde gegevens over het verloop van de primaire processen worden bijgehouden;
- een database voor beslissingsondersteuning waarin naast individuele gegevens en gegevens afkomstig uit de omgeving via bijvoorbeeld publieke databases, geaggregeerde gegevens uit primaire processen worden vastgelegd.

Hiervoor zijn verschillende, meest technische, argumenten aan te geven:

1. De efficiëntie van bestaande database management systemen en met name relationele dbms is (nog steeds) een technologisch probleem als geprobeerd wordt fysiek één geïntegreerde database toe te passen.
2. Persoonlijk computergebruik vraagt nog een andere hardware- of softwareomgeving dan geautomatiseerde gegevensverwerking.
3. Gegevens in de database voor beslissingsondersteuning vertonen niet dezelfde nauwkeurigheid als die in de primaire database. Daarnaast is vervuiling niet denkbeeldig.

De database voor beslissingsondersteuning omvat vaak gegevens die uit basisgegevens geaggregeerd worden, bijvoorbeeld over de tijd of over produktenmerken. Naast deze technische scheiding komt uit verschillende studies naar voren dat er vraagtekens geplaatst kunnen worden achter de validiteit van managementinformatie die door aggregatie tot stand komt, zie bijvoorbeeld Reuijl [1983], Sol [1982]. Voor diverse beslissingen is het juiste niveau van aggregatie vaak niet op voorhand aan te geven. Geaggregeerde gegevens geven niet altijd aanknopingspunten voor het onderkennen van verbanden, waarmee een betere afstemming van beslissingsprocessen kan worden gerealiseerd. Toch worden bij vele beslissingen juist geaggregeerde gegevens gebruikt. Men kan in deze gevallen echter een oplossing vinden door op simulatie gebaseerde beslissingsondersteunende systemen te hanteren, die mogelijkheden bieden om op detailniveau via disaggregatie uitspraken te doen.

Werkwijze

Met betrekking tot strategieën voor het ontwerpen van informatiesystemen staat prototyping sterk in de belangstelling. In het algemeen kan men van prototyping zeggen dat het zich richt op het snel creëren van werkende systemen. Daarmee kan worden nagegaan of het prototype een valide model is van het RS of het IS, dat aan de functionele eisen en wensen in de probleemsituatie beantwoordt. Prototyping kan worden gehanteerd om

- een beter begrip van een toepassingsgebied te krijgen;
- informatiebehoeften juist en volledig te specificeren;
- gebruikers-interfaces te ontwikkelen;
- (voor) productie testen uit te voeren.

Op basis van diverse praktijkstudies vat ik mijn ervaringen met prototyping samen in de paradox van INCREMENTEEL ONTWERPEN EN CONVERGENTIE.

Ik concludeer, zie Sol [1984], dat bij prototyping

- verificatie en validatie van het kenmodel niet altijd de benodigde aandacht krijgen doordat het snel hard maken van een dynamisch modelstelsel de overhand heeft;
- niet iedere organisatie rijp is voor een wegwerp aanpak inherent aan prototyping. Met name geldt dit voor bureaucratisch getinte organisaties. Ook druist dit vaak in tegen de belevingswereld van 'traditionele' automatiseerders. Het over boord gooien van een oplossing waar tijd en mankracht in is gestoken, is niet altijd gemakkelijk voor betrokkenen. Een organisatie moet voorbereid zijn op het naar boven komen van conflicten en bereid zijn tot het zoeken naar oplossingen hiervoor. Dit betekent dat (top)management een prototyping aanpak zwaar moet (onder)steunen;
- de overgang van een kenmodel naar een te maken doelsysteem nauwelijks uit de verf komt;
- de oplossing snel als definitief door de organisatie wordt overgenomen zonder dat verder geëxperimenteerd wordt;
- de effectiviteit meer centraal staat dan de efficiëntie;
- modellen als vereenvoudiging van de uitgangssituatie gehanteerd worden, waarbij het totaalbeeld gauw uit het oog wordt verloren. Het gevaar van de-

ze 'tunnelvisie' kan mogelijkwijs beperkt worden door te werken vanuit een goed informatiseringsplan voor een organisatie.

Kortom, incrementeel ontwerpen staat niet garant voor een snelle convergentie naar een acceptabel doelsysteem.

Ik merk op dat er zeer weinig gegevens beschikbaar zijn over de middelen die aangewend worden in de verschillende activiteiten in het ontwikkelingstraject in relatie tot de kwaliteit van het eindproduct. Dit geldt zeker voor prototyping. Veel gegevens over de efficiëntie en de effectiviteit van diverse methoden ontbreken nog. Er zijn aanwijzingen dat een strak hanteren van methodieken zeker niet altijd tot produktiviteitsverhoging aanleiding geeft. Daarnaast moet men constateren dat aan methoden voor de meting van produktiviteit, zoals b.v. een functiepuntenanalysemethode, zie Rudolph [1983], slechts beperkt gewicht kan worden toegekend. Immers, deze methoden dekken slechts een gedeelte van het ontwerptraject, nl. vanaf het moment dat een globaal doelsysteem duidelijk gedefinieerd is.

Beheerswijze

Wat het aspect van projectbesturing betreft valt het op dat gereedschap voor volgordeplanning en capaciteitsplanning van automatiseringsprojecten nog weinig ingevoerd is. In Sol [1982] is geconstateerd dat in slecht gestructureerde problemen een incrementele aanpak niet tot langere doorlooptijden hoeft te leiden dan een lineaire of iteratieve aanpak. De techniek van simulatie van stochastische netwerken is ook bruikbaar om bij de start van een project tot een voorlopige doorlooptijdschatting te komen.

In dit verband wijs ik erop dat een lineaire aanpak andere afspraken vereist dan een iteratieve of een incrementele. Een projectvorm volgens een incrementele strategie vertoont veel overeenkomst met een research en development project, en kan met soortgelijke contracten bestuurd worden.

Tot slot merk ik op dat in methodieken soms veel nadruk gelegd wordt op controle-lijsten waaraan projectrapporten en documenten moeten voldoen. Alhoewel niet genoeg aandacht gegeven kan worden aan een goede en volledige project- en programmadocumentatie, heb ik wel de indruk dat deze aandacht soms ten koste gaat van de kwaliteitscontrole en -beheersing.

4 ONTWERPEN EN EXPERTISE

Ontwerpen van informatiesystemen is spelen met modellen van het RS en het IS om problemen te ordenen, te formuleren, op te lossen en te implementeren. Het creatief hanteren van methodieken, methoden en ontwerpfilosofieën speelt een belangrijke rol in dit proces. Een ontwerper van informatiesystemen heeft te maken met:

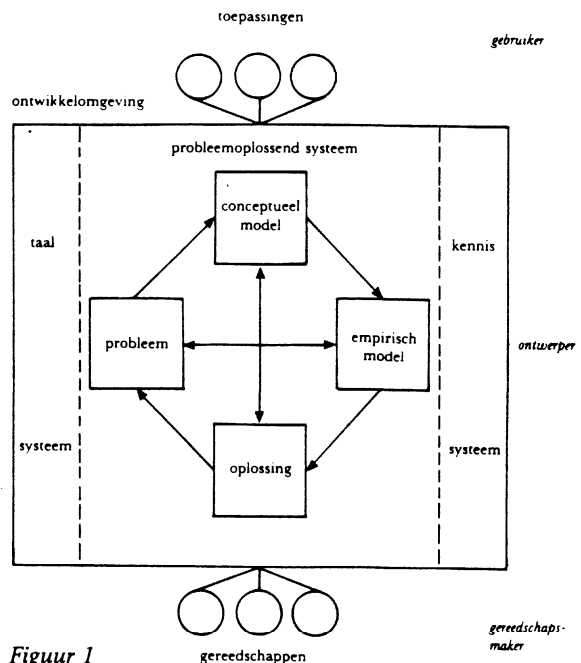
- de keuze van een Weltanschauung en een ontwerpfilosofie, die vertaald moeten worden in een aantal bouwstenen en gereedschappen;
- de keuze van een methodiek in een specifieke probleemsituatie, die aangeeft in welke volgorde de activiteiten in het ontwerpproces worden doorlopen en welke beschrijvingswijzen mogelijk zijn om de huidige en de gewenste situatie weer te geven;
- de keuze uit methoden voor de analyse van de bestaande situatie en het genereren van nieuwe al-

ternatieven, waarbij de organisatorische mogelijkheden in het oog gehouden moeten worden.

De architectuurfunctie van de informatiekundige wordt steeds belangrijker én complexer. Immers, het aantal configuratiemogelijkheden van apparatuur en programmatuur neemt sterk toe. Daarnaast raken informatiesystemen door de voortschrijdende informatisering dieper verstrengeld in de organisatie. Kijkend naar ervaringen rond informatiesystemen constateer ik dat er nog weinig generaliseerbare expertise bestaat en dat deze bovendien nog weinig toegepast wordt. Er zijn weinig empirische gegevens beschikbaar over het ontwerpproces op zich en over de kwaliteit van het eindproduct. Dit geldt in het bijzonder voor de 'contingency method' en voor prototyping. Het opbouwen en vastleggen van de aanwezige expertise bij automatiseringsprojecten wordt veelal verwaarloosd.

De jonge leeftijd van het vakgebied informatiesystemen onderschrijft eens te meer de noodzaak om én voor praktijkgericht onderzoek én voor theorievormend onderzoek een harde kern van een onderzoekprogramma met een aantal beschermende ringen aan te geven, in navolging van Lakatos [1970]. Voor een uitwerking hiervan verwijs ik naar Bosman en Sol [1984a, 1984b]. De harde kern van dit programma gaat allereerst uit van de optiek dat gebruikers en beslissers beperkt rationeel handelen. Daarnaast legt de modelleringswijze de nadruk op het formuleren van hypothesen in een probleemsituatie, het maken van een conceptueel model op basis hiervan, het verifiëren en valideren van deze beschrijving, het genereren van alternatieve oplossingen aan de hand van dit vertrekpunt en de implementatie van de gekozen oplossing.

De beschermende ringen zijn opgebouwd rond het concept van een ONTWIKKELOMGEVING voor informatiesystemen. Door Sprague [1980] is een raamwerk voor het ontwikkelen van BOS geïntroduceerd, met als belangrijke componenten een applicatie-generator, gereedschap en specifieke toepassingen. Bonczek et al. [1981] geven hieraan een uitbreiding door de concepten



Figuur 1

van een database, een modellenbank en een dialoogsysteem te vervangen door die van een taalsysteem, een kennissysteem en een probleemoplossend systeem. Echter, het probleemoplossend systeem wordt door hen niet ingevuld. Mitroff et al. [1974] hebben een kader geschetst, waarin verschillende aanpakken bij probleemoplossen kunnen worden aangegeven. Deze bijdragen zijn in figuur 1 geïntegreerd. Men kan diverse verschijningsvormen van informatiesystemen hierin onderbrengen. Wanneer men bijvoorbeeld het kennissysteem als een bestand van gegevens, feiten en regels opvat en het probleemoplossend systeem als een 'inferentie machine', dan wordt het schema specifiek voor expertsystemen. In verschillende studies, zie o.m. Sol [1982], heb ik laten zien dat het mogelijk is zo'n ontwikkelomgeving voor informatiesystemen te hanteren door een specifieke invulling te geven aan een beschrijvingsapparaat (een taalsysteem), een beschrijving van een toepassingsgebied (een kennisbank) en aanwijzingen hoe een conceptueel model en een empirisch model ontwikkeld kunnen worden, zowel beschrijvend als voorschrijvend (een probleemoplossend systeem).

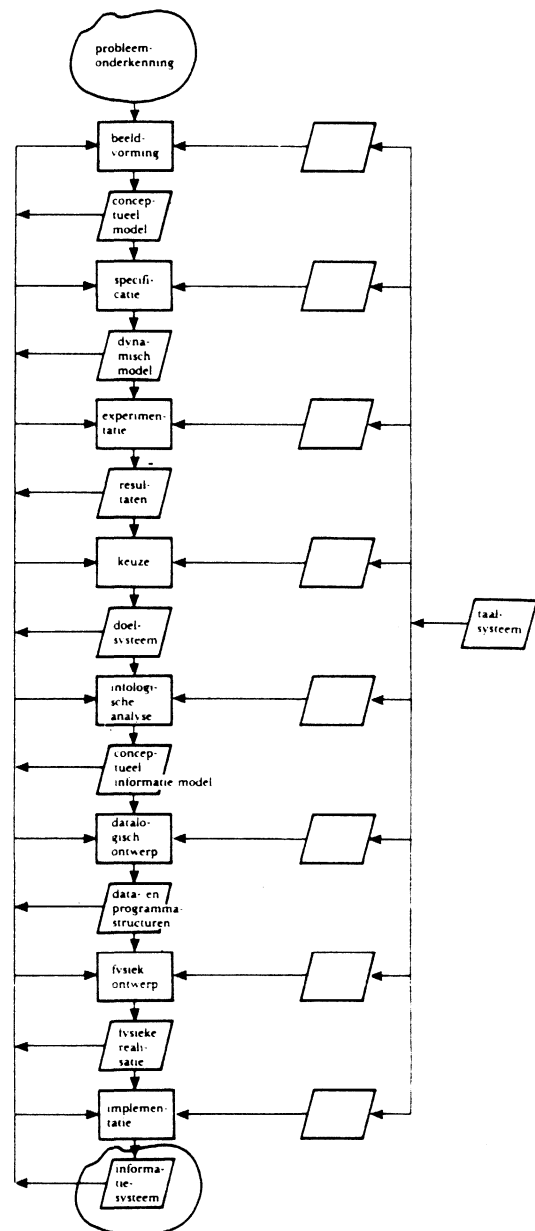
Een aanpak is dan de volgende:

1. Maak een beschrijving van het objectsysteem, door proces-objecten en data-objecten vast te leggen.
2. Probeer vervolgens taken en beslissingsregels te achterhalen en maak hiermee een dynamisch model.
3. Analyseer de bestaande situatie met behulp van dit simulatiemodel en formuleer het probleem.
4. Creëer op basis van de kennis opgedaan met dit kenmodel én op basis van een ontwerpfilosofie, alternatieve oplossingen en test deze als prototype.
5. Kijk hoe het prototype efficiënt in een informatieverwerkend systeem geïmplementeerd kan worden.

In het licht van eerdere opmerkingen geef ik de voorkeur aan een taalsysteem, dat uitgaat van een object als combinatie van een entiteit met acties, cf. Bubenko [1983], Cox [1984], Olivé [1982], Sol [1982]. Ieder object bestaat dan in een eigen omgeving of domein. De actie-specificatie is zowel procedureel als non-procedureel. Verschillende coördinatievormen tussen objecten zijn denkbaar. Ter vermijding van mogelijke misverstanden merk ik op dat een dergelijk taalsysteem niet haaks op gebruiksvriendelijkheid of specificatiegemak hoeft te staan. Immers, uit de wereld van simulatie van systemen en interactieve modelbouw is bekend dat gebruiker en ontwerper interactief een conceptueel en empirisch model kunnen vastleggen, dat vervolgens in een formele taal wordt gerepresenteerd, c.f. Birtwistle [1984].

Een dergelijk ontwikkelsysteem kan gerealiseerd worden als een elektronische werkplek met faciliteiten voor tekstverwerking, documentatie, grafische weergave, dynamische systeembeschrijving en simulatie. Bij voorkeur moet dan gewerkt worden op een werkplek met meervoudige vensterfaciliteiten.

Een strategie zou dan kunnen zijn om per afdeling of groep van individuen een dynamische objectbeschrijving op een apart ontwikkelsysteem te realiseren als prototype. Vervolgens kunnen dan voor de organisatie of een gedeelte ervan de verschillende prototypes gecoördineerd worden. Een methodologisch probleem is echter om voor de afzonderlijke beschrijvingen én voor de



Figuur 2

coördinerende beschrijving snel modellen van doelsystemen te ontwikkelen. Wellicht kunnen specifieke systemen voor ondersteuning van deze expertfuncties uitkomst gaan bieden.

Ook is hierbij te denken aan methodieken om een bestaande organisatie met de daarin aanwezige informatiesystemen door te lichten. Een analyse van Critical Success Factors, zie Alloway [1980] en Rockart [1982], zou handvaten kunnen bieden om tot een betere probleemherkenning in de 'understanding stage' te komen.

In deze ontwikkelomgeving is een belangrijke plaats ingeruimd voor expertsystemen ter ondersteuning van het ontwerpproces. Ik denk daarbij aan de volgende verschijningsvormen (zie figuur 2).

1. Een expertstelsysteem ter ondersteuning van de constructie van beschrijvende conceptuele modellen en

empirische modellen van de bestaande situatie gebaseerd op 'application domain models'. In een dergelijk application domain zijn basisbegrippen en basisprocessen van een bedrijfstak of toepassingsgebied als prototype – in letterlijke betekenis – beschreven.

2. Een expertsysteem ter ondersteuning van de constructie van een simulatiemodel door handvaten voor verificatie, validatie en gevoeligheidsanalyse aan te reiken.
3. Een expertsysteem ter vergemakkelijking van de constructie van een proefopzet. Met nadruk zij opgemerkt dat 'automation of thinking in design' beperkt is, omdat deze hulp voor experimentatie en statistische analyses naar mijn mening slechts ten dele te automatiseren is.
4. Expertsystemen die handreikingen kunnen leveren voor de overgang van kenmodel naar doelsysteem. Hier speelt echter de vraag in hoeverre zulke expertsystemen generaliseerbaar zijn voor een specifiek toepassingsgebied en hoe deze expertise te verkrijgen is.
5. Een expertsysteem als kennisbank of consumentengids met mogelijke apparatuur- en programmaatuuurconfiguraties, die aan het maakmodel als uitkomst van het systemologische probleem, kunnen voldoen. Hiermee kan dan snel nagegaan worden of een systemologische oplossing, infologisch, datologisch en technologisch haalbaar is en tegen welke kosten.
6. Het lijkt mogelijk een kennisbank op te bouwen waarin empirisch materiaal over het verloop van automatiseringsprojecten is opgenomen, zoals kenmerken van de probleemsituatie, kenmerken van het produkt, en cijfers per activiteit, b.v. projectduur en gebruikte middelen. Hieruit kunnen aanwijzingen voor de projectfasering geput worden.

Eén relativerende opmerking is echter rond deze expertsystemen op zijn plaats, en wel de valkuil van de expertise. De genoemde expertsystemen zijn bedoeld om tot betere ontwerpen en implementaties in toekomstige situaties te komen. De expertise is en blijft echter gebaseerd op ervaringen uit het verleden. Een kritiekloos doortrekken naar de toekomst lijkt mij niet ongevaarlijk. Juist de combinatie van expertise en creativiteit moet informatiesystemen opleveren die de toets van falsificatie kunnen doorstaan.

Het interessante, en nieuwe van deze aanpak is, dat uitgegaan wordt van parallel bestaande en opererende objecten. Na beschrijving en dynamische analyse van individuele objecten kunnen deze als prototypes van informatie-systemen worden bevroren. Vervolgens wordt nagegaan welke conventionele architectuur een efficiënte implementatie mogelijk kan maken.

Een andere weg ligt echter in de komende jaren meer voor de hand. De beschrijving in parallelle prototypes wordt vertaald naar een (volledig) parallelle architectuur. De realisatie hiervan hangt af van technologische ontwikkelingen en economische mogelijkheden.

5 ONDERWIJS EN ONDERZOEK

Ideeën over het vakgebied Informatiesystemen kunnen mooi zijn, maar belangrijker zijn praktische uitvoerbaar-

heid en concretisering in een onderwijsprogramma en een onderzoekprogramma. Laat ik allereerst een invulling van een onderzoekprogramma schetsen, al was het alleen maar om aan te geven dat naar mijn mening onderzoek een noodzakelijke voorwaarde is om tot goed onderwijs te komen.

In het onderzoekprogramma dat voor de Werkenheid Informatiesystemen van de TH Delft geformuleerd is, wordt een viertal hoofdgebieden onderscheiden:

- 1 Ontwerpen van complexe informatiesystemen.
- 2 Effectiviteit van informatiesystemen.
- 3 Methodieken voor strategische planning voor de informatievoorziening.
- 4 Ontwikkeling van expertsystemen.

5.1 Ontwerpen van complexe informatiesystemen

Een verbetering van een ontwerp aanpak voor complexe informatiesystemen wordt nagestreefd enerzijds langs de weg van methodieken en hulpmiddelen, anderzijds empirisch door voor verschillende klassen van probleemsituaties informatiesystemen te ontwikkelen en op een inductieve wijze tot verbeteringen te komen in relatie tot te hanteren ontwerptheorieën:

- concretisering van ontwikkelomgevingen die van een object-gerichte specificatie uitgaan. Hierbij gaat het om:
 - de invulling van een taalsysteem,
 - de invulling van gereedschap in elk der activiteiten in het ontwerpproces, en in het bijzonder de rol van expertsystemen hierin,
- de constructie van werkstations waarop ontwikkelomgevingen gehanteerd kunnen worden,
- de uitwerking van een prototyping aanpak;
- empirische toetsing van ontwikkelomgevingen voor verschillende klassen van probleemsituaties.

5.2 Effectiviteit van informatiesystemen

Dit heeft betrekking op gegevensverwerkende systemen, managementinformatiesystemen, en beslissingsondersteunende systemen. Specifieke vraagstellingen zijn hier:

- de invulling van ontwikkelomgevingen voor beslissingsondersteunende systemen en de bepaling van de effectiviteit hiervan met behulp van interactieve bedrijfsspelen;
- de bepaling van de effectiviteit van ontwikkelomgevingen, waaronder methodieken, hulpmiddelen en mankracht, in relatie tot kenmerken van probleemsituaties, voor de vastlegging waarvan een diagnose apparaat wordt ontwikkeld;
- richtlijnen om gegevensverzamelingen t.b.v. beslissingsondersteuning te aggregeren op basis van een classificatie van probleemsituaties;
- effectiviteit van informatiesystemen afhankelijk van de wijze waarop in organisaties de complexiteit wordt benaderd door een oplossing voor het coördinatieprobleem te kiezen.

5.3 Methodieken voor de strategische planning voor de informatievoorziening

Centraal staat hier een verbetering van strategische planningsmethodieken voor de informatievoorziening in maatschappelijke sectoren en grote gedifferentieerde en gespreide organisaties.

5.4 Ontwikkeling van expertsystemen

Deze activiteit maakt deel uit van het voorgestelde onderzoekprogramma Kunstmatige Intelligentie en richt zich op de ontwikkeling van werkende expertsystemen in verschillende toepassingsgebieden. Op inductieve wijze wordt getracht inzicht te verwerven in de toepasbaarheid van expertsystemen en de te hanteren ontwerpaanpak.

De ideeën over de invulling van een onderwijsprogramma rond informatiesystemen zijn elders uitvoerig beschreven, zie Bosman en Sol [1983]. Ik ben van mening dat een programma op dit vakgebied niet mag bestaan uit een verzameling naast elkaar staande vakken, waarbij de integratie aan de student wordt overgelaten. Het is essentieel een integratievoertuig te kiezen dat als bindmiddel kan fungeren. In de afstudeerrichting Informatiesystemen binnen de TH Delft is hiervoor gekozen. Zoals gesteld, wordt het systemisch ontwerpprobleem steeds belangrijker. Dit betekent dat het maken van valide, herkenbare beschrijvingen van het objectsysteem in een probleemsituatie meer aandacht moet krijgen. Voor het maken van plaatjes of conceptuele modellen is het nadenken over reëel systeem en informatiesysteem belangrijk. Ik zou echter nog een stap verder willen gaan en in het onderwijs de nadruk willen leggen op een interdisciplinaire beschrijving van een objectsysteem als dynamisch empirisch model. Van daaruit kan men dan komen tot een ontwerp voor het oplossen van de probleemsituatie en voor het informatiesysteem dat daarvoor nodig is. De computer wordt daarbij dus niet gezien als vervanging van het menselijk oordeel, maar als hulpmiddel ter vergroting van het probleemoplossend vermogen van individuen, groepen en organisaties.

Teneinde het onderzoekprogramma en het onderwijsprogramma direct te laten aansluiten bij praktijkproblemen rond informatiesystemen in organisaties stel ik voor aan deze Hogeschool een Centrum voor Informatiesystemen op te richten als laboratorium voor het ontwerpen, ontwikkelen en implementeren van informatiesystemen in kleine én grote organisaties.

Het Centrum kent een tweeledig doel:

a. Opleiding

1. Het bieden van een efficiënte trainingsomgeving voor 3e en 4e jaars (Informatica) studenten in het ontwerpen van informatiesystemen, zowel geautomatiseerde gegevensverwerkende systemen als beslissingsondersteunende systemen.
2. Het bijscholen van externe deelnemers in het doorlichten van informatiesystemen, in het formuleren van informatiebeleid en in het concretiseren van informatieplannen, in methoden en technieken voor het ontwerpen van informatiesystemen.

b. Praktijkgericht onderzoek, binnen het genoemde onderzoekprogramma.

Voor beide doelgroepen worden korte inleidende seminars verzorgd. Vervolgens worden verschillende reële cases ingebracht door externe deelnemers. Deze cases worden door studenten-teams uitgewerkt met externe deelnemers in verschillende rollen, bijvoorbeeld die van gebruiker of instructeur. Daarnaast worden verschillende bedrijfsspelen gehanteerd, waardoor door diverse teams informatiesystemen ontwikkeld worden.

Als hulpmiddelen zijn verschillende ontwerpomgevin-

gen, zowel op mainframe als op microcomputers, in het Centrum beschikbaar. Deze omgevingen omvatten methodieken, hulpmiddelen voor projectplanning, een workbench voor informatie-analyse en systeemontwerp, werkstations voor interactief ontwerp en documentatie, simulatie hulpmiddelen, vraagtaalen, financiële modelleringspakketten, etc.

Ik hoop dat de realisatie van deze voorstellen zal aantonen, dat, zo er sprake is van een tegenstelling theorie-praktijk, dit een schijn tegenstelling is.

6 LITERATUUR

- Alloway, R. M., *Defining succes for data processing: a practical approach to strategic planning for the DP-Department*, MIT, CISR 52, Mass, 1980.
- Bemelmans, Th. M. A., Empirisch onderzoek over informatiesystemen: Verslag van een conferentie, *Informatie*, jrg. 25, nr. 2, (februari 1983).
- Birtwistle, G. M., The Jade Project, *Proceedings of the SCS Simulation Conference on Strongly Typed Languages*, San Diego, 1984.
- Bonczek, R. H. et al., *Foundations of Decision Support Systems*, Academic Press, New York, 1981.
- Bosman, A., *Een metatheorie over het gedrag van organisaties*, Stenfert Kroese, Leiden, 1977.
- Bosman, A., Sol, H. G., Bestuurlijke informatiekunde in het wetenschappelijk onderwijs, *Informatie*, jrg. 25, nr. 6, (juni 1983).
- Bosman, A., Sol, H. G., Systemen voor management ondersteuning, in: Bosman, A., Sol, H. G., Oonincx, J. A. M. (eds.), *Ontwikkelingen rond informatiesystemen*, Samsom, 1984a.
- Bosman, A., Sol, H. G., *Knowledge representation and information systems Design*, Proceedings IFIP WG 8.1 Working Conference, Durham, 1984b.
- Bubenko, J. A., et al., A declarative approach to conceptual information modeling, in: Olle, T. W., Sol, H. G., Verrijn Stuart, A. A. (eds.), *Information Systems Design Methodologies: A Comparative Review*, North-Holland, Amsterdam, 1982.
- Budde, R., Sylla, K-H, From application domain modeling to target system, in: Budde, R. et al., (eds.), *Approaches to prototyping*, Springer, Berlin, 1984.
- Burch, J. C., Strater, F. R., *Information Systems: Theory and Practice*, Hammliton Publishing Company, Santa Barbara, 1974.
- Cox, B. J., Message/object programming: An evolutionary change in programming technology, *IEEE Software*, January, 1984.
- Davis, G. B., McKeen, J. D. en Naumann, J. D., Determining information requirements: A contingency method for selection of a requirements assurance strategy, *The Journal of Systems and Software*, 1, 1980.
- Falkenberg, E. et al., Feature analysis of ACM/PCM, CIAM, ISAC and NIAM, in: Olle, T. W., Sol, H. G., Tully, C. J. (eds.), *Information Systems Design Methodologies: A Feature Analysis*, North-Holland, Amsterdam, 1983.
- Kensing, F., Property determination by prototyping, in: Budde, R. et al., (eds.), *Approaches to prototyping*, Springer, Berlin, 1984.
- Lakatos, I., Falsification and methodology of scientific research programmes, in: Lakatos, I., Musgrave, A. (eds.), *Criticism and the growth of knowledge*, University Press, Cambridge, 1970.
- Langefors, B., *Theoretical analysis of information systems*, Student Litteratur, Lund, 1966.
- Lith, P. van, D2S2, *Informatie*, Jrg 24, nr. 11 (november 1982) Vervolg literatuur op pag. 196

informatiebehoefte van de gebruiker, correct en volledig is.

Aan de andere kant is het duidelijk dat normalisatie op tenminste twee punten te kort schiet:

- het kent geen specialisaties c.q. generalisaties,
- het geeft niet aan wanneer het 'automatisme' van splitsing van relaties moet worden doorbroken.

6 REFERENTIES

1. E. F. Codd, A relational model of data for large shared data banks. *CACM*, 13 (1970), pp. 377-387.
2. E. F. Codd, *Further normalization of the database rela-*

tional model, Courant Computer Science Symposia 6. Database Systems, Prentice-Hall, NJ, 1971, pp. 65-98.

3. E. F. Codd, Extending the database relational model to capture more meaning, *Transactions on Database Systems* vol. 4 (1979) pp. 397-434.
4. C. J. Date, *An introduction to database systems*, 3rd edition, Addison-Wesley.
5. J. M. Smith en D. C. P. Smith, Database abstractions: aggregation and generalization, *Transactions on Database Systems*, vol. 2, (1977) pp. 105-13

Met dank aan ir. J. H. Pels voor zijn kritisch commentaar.

JOURNAL OF MANAGEMENT INFORMATION SYSTEMS

Inmiddels zijn de eerste afleveringen van dit Amerikaanse tijdschrift (met een internationale Board of Editors, bestaande uit Japanners, Europeanen en Amerikanen) verschenen, en hoewel het niet onze gewoonte is hier elk nieuw vakblad te bespreken (ze komen en gaan vaak met de seizoenen) willen wij toch graag een uitzondering maken voor JMIS; al was het alleen maar omdat de redactionele aandacht van JMIS met zekere beperkingen gelijk is aan die van INFORMATIE.

Zoals de titel al aanduidt houdt dit (kwartaal)blad zich bezig met management information systems, een betiteling die in Nederland vaak geassocieerd wordt met het voorvoegsel mis- (als in misgreep). Toch, zoals de editor-in-chief Vladimir Zwass (professor aan de Fairleigh Dickinson University, New Jersey) stelt in de inleiding van het eerste (zomer 85) nummer, in de VS 'the outlines of MIS as an academic discipline have been formed, ... and a growing body of research has been accumulated. The express purpose of JMIS, is to serve as a medium of communication between MIS researchers and practitioners.' Een greep uit de onderwerpen die aan de orde zouden moeten komen: de integratie van informatiebeleid en ondernemingsbeleid, decision support systems, systeem-

ontwikkelingsmethoden, AI, het management van de technologische evolutie, de integratie van personal computing en office automation in MIS, de participatie van de eindgebruikers, beheer en beveiliging etc.

Aan deze redactionele plannen is in de twee eerste nummers met de volgende artikelen gevolg gegeven:

- Eliciting decision-makers information requirements
- Capturing more meaning in databases
- Designing financial information systems
- A database architecture for business
- A manufacturing resource planning IS
- Writing simple program generators
- Organisation size and information system use
- Information systems semantics
- Information systems planning and development
- Fighting information pollution
- DSS mechanisms for judgmental flexibility

Zoals eerder al geïmpliceerd zou deze serie artikelen ook in INFORMATIE gepubliceerd kunnen zijn; dat geldt over het algemeen ook voor de inhoudelijke kwaliteit. JMIS verschijnt 4x per jaar en kost \$ 75 voor 'institutions', \$ 35 voor 'individuals'. Het adres: JMIS, M. E. Sharpe Inc 80 Business Park Drive, Armonk, New York 10504.

Vervolg literatuur art. Kennis en ervaring rond het ontwerpen van informatiesystemen, pag. 192

- Lundeberg, M., The ISAC approach, in: Olle, T. W., Sol, H. G., Verrijn Stuart, A. A. (eds.), *Information Systems Design Methodologies: A Comparative Review*, North-Holland, Amsterdam, 1982.
- Mitroff, I. I., Betz, F., Pondy, I. R., en Sagasti, F., On managing science in the systems age: Two schemes for the study of science as a whole systems phenomenon, *TIMS Interfaces*, Vol. 4, nr. 3, 1974.
- Mumford, E., Experiences with socio-technical design, in: Bemelmans, Th. M. A. (ed.), *Beyond productivity: Information systems for organizational effectiveness*, North-Holland, Amsterdam, 1984.
- Nygaard, K. et al., *Introduction to the Beta programming language*, NCC, Oslo, 1978.
- Olivé, A., DADES, in: Olle, T. W., Sol, H. G., Verrijn Stuart, A. A., *Information Systems Design Methodologies: A Comparative Review*, North-Holland, Amsterdam, 1982.
- Reuijl, J. C., *On the determination of advertising effectiveness, an empirical study of the German cigarette market*, Stenfert Kroese, Leiden, 1983.
- Rockart, J. F., *The changing role of the information systems executive: A critical success factors perspective*, Proceedings Third International Conference on Information Systems, Ann Arbor, Michigan, 1982.
- Rudolph, E. E., *Productivity in computer application development*, W. P. no 9, University of Auckland, 1983.
- Sebus, G. M. W., Business Systems Planning, *Informatie*, jrg 23, nr. 3, (maart 1981)
- Sol, H. G., *Simulation in information systems development*, Dissertatie Rijksuniversiteit Groningen, 1982.
- Sol, H. G., *Processes and tools for decision support: Inferences for future developments*, North-Holland, Amsterdam, 1983.
- Sol, H. G., Prototyping: A methodological assessment, in: Budde, R. et al. (eds.), *Approaches to prototyping*, Springer Berlin, 1984.
- Sol, H. G., Ijpelaar, D. P. N. M., Methodieken voor informatiesysteemontwikkeling: De volgende stap in een vergelijking, *Informatie*, Jrg. 25, nr. 9, (september 1983).
- Sprague, R. H., A framework for research on Decision Support Systems, in: Fick, G., Sprague, R. H. (eds.), *Decision Support Systems: Issues and challenges*, Pergamon Press, Oxford, 1980.
- Verrijn Stuart, A. A., *Kwantitatieve aspecten van informatiesystemen*, Samsom, Alphen aan den Rijn, 1976.
- Welke, R. J., Current information system analysis and design approaches, in: Buckingham, R. A. (ed.), *Education and large information systems*, North-Holland, Amsterdam, 1977.

DE INFOMOD-BENADERING VAN INFORMATIEMODELLERING

door J. J. van Griethuysen en dr. D. A. Jardine

INFOMOD is een benaderingswijze en taal voor het specificeren van een informatiemodel. Een informatiemodel modelleert in de eerste plaats die dingen, zaken, situaties en gebeurtenissen, waarover informatie verlangd wordt. Voor het bewaren en doorgeven van deze informatie kan een (geautomatiseerd) informatiesysteem gebruikt worden. Derhalve biedt INFOMOD ook mogelijkheden voor het specificeren van uitwisseling van deze informatie tussen gebruikers van het informatiesysteem en dit systeem zelf. Een dergelijk informatiemodel behelst een overeenstemming tussen de gebruikers waarover de informatie handelt, m.a.w. wat deze betekent. Op deze wijze vormt het ook de specificatie welke informatie uit het informatiesysteem verkregen kan worden. INFOMOD is een ontwikkeling van N.V. Philips' Gloeilampenfabrieken en wordt daar op tal van plaatsen toegepast.

1 KARAKTERISERING

1.1 Inleiding

In dit artikel wordt een verkorte uiteenzetting gegeven van een methode en formele taal voor het beschrijven van informatie en de uitwisseling hiervan. Deze methode en taal kunnen gebruikt worden voor het opstellen van nauwkeurige informatiemodellen welke kunnen dienen voor de constructie van op computers verwezenlijkte informatiesystemen. De methode en de bijbehorende taal wordt INFOMOD genoemd, afgeleid van INforMatie MOdel. Zij kunnen zowel voor specificatie van op bedrijfsorganisaties gerichte systemen gebruikt worden, als ook voor toepassing in het CAD/CAM-gebied. Hoewel de theorie en toepassingen beschreven in dit artikel in de eerste plaats gericht zijn op systemen, waarin computers gebruikt worden, zijn zij veel algemener geldend.

INFOMOD is geschikt voor verscheidene aspecten van informatieverwerking. Om er enkele te noemen:

- analyse van informatie behoeften;
- specificatie van eisen voor informatieverwerking te stellen aan informatiesystemen;
- formulering van regels voor de beheersing van de informatie en voor het voorkomen van tegenspraak in de in het systeem beschikbare informatie;
- beschrijving van de betekenis van data.

De elementen, waaruit een informatiemodel wordt opgebouwd, dienen gefundeerd te zijn op een hechte theoretische basis. INFOMOD is gebaseerd op de principes van formele logica en verzamelingenleer. Hiermee sluit het, evenals bv. NIAM van prof. Nijssen [10] en RIDL van Control Data [9], aan op de uitgangspunten voor informatiemodellering zoals door de ISO-werkgroep ISO TC97/SC5/WG3 geformuleerd in het rapport 'Concepts and Terminology for the Conceptual Schema and the Information Base' [5].

De elementaire constructies van deze theoretische basis zijn echter niet zo 'gebruiker-vriendelijk' voor de meesten van ons. Daarom omvat INFOMOD een formele taal, die dicht ligt bij natuurlijke taal, en daarom wellicht beter geschikt is voor degenen wier interesse meer gericht is op de formulering van een (informatie)probleem, dan op de formulering van de theoretische grondslag van

de methode. De taal is echter zo ontworpen, dat zij ook op een computer verwerkt kan worden. Dit kan eveneens gezegd worden van RIDL.

1.2 Enkele uitgangspunten voor informatiesysteem-ontwerp

(Software) System Engineering, waarbinnen INFOMOD een plaats kan vinden, gaat uit van de Three Level Approach [1, 5]. Dit betekent, dat een duidelijke scheiding gelegd wordt tussen:

1. Wat is er nodig.
2. Hoe ziet dit er voor de gebruiker uit.
3. Hoe wordt dit door de hulpmiddelen (computersystemen) verwezenlijkt.

Hierdoor kunnen eisenstelling en ontwerpbeslissingen voor ontwerp en constructie van informatiesystemen gestructureerd genomen worden. B.v. de eisen en behoefte aan informatie kunnen door de eindgebruikers van de informatie geformuleerd worden zonder deze met praktische, op de (ad hoc) toe te passen computertechnieken gerichte, beslissingen te vertroebelen. Hierna kunnen deze eindgebruikers hun eisen en wensen formuleren voor de vormen, waarin deze informatie verwerkt en uitgewisseld moet worden. In de praktijk zal dit zeker in overleg met bouwers van informatiesystemen plaats vinden. Uitgaande van een dergelijke specificatie, kan dan een optimale constructie ontworpen worden gebaseerd op de voor handen zijnde technieken en mogelijkheden van de huidige data processing.

Ook deskundigen als James Martin stellen terecht, dat specificaties voor informatie-uitwisseling en verwerking in principe onafhankelijk dienen te zijn van (huidige) implementatie-technieken [8]. Hierbij adviseren zij o.m.:

1. Logisch (dwz. onafhankelijk van computerrepresentatie) georiënteerde 'user views'.
2. Non-procedurele specificaties.
3. Specificatietalen ondersteund door (4e generatie) tools, die gericht zijn op (in de nabije toekomst) automatisering van declaratie en codegeneratie, waar mogelijk.

Tegenwoordig wordt het gehele analyse- en ontwerp-proces van informatiesystemen in een aantal fasen opgedeeld, welke ruwweg als volgt gekarakteriseerd kunnen worden:

- analyse en specificatie:
- (logisch en technisch) ontwerp:
- implementatie en test.

Hierbij richt de analyse en specificatie zich vooral op zaken als:

- Specificatie wat voor informatie over wat voor zaken nodig is voor welke bedrijfsfuncties op welke momenten.
- Hoe de bedrijfsfuncties (de gebruikers) deze informatie en de uitwisseling hiervan via een informatiesysteem gepresenteerd willen zien, onderverdeeld in eisen en wensen.

Dergelijke specificaties zullen in eerste instantie sterk georiënteerd zijn op individuele bedrijfsfuncties (specifieke 'user views'). Deze kunnen van zeer globale aard zijn, maar ook op zeer gedetailleerde zaken gericht zijn. Om te komen tot een geïntegreerd systeem zullen deze individuele specificaties worden samengevoegd, hetzij vanaf het begin, hetzij als (latere) toevoegingen aan een bestaand systeem.

Hierna komt het logisch ontwerp van het informatiesysteem. Doelstellingen hierbij zijn:

- Vaststelling in overleg met de verantwoordelijke en betrokken gebruikers hoe de informatieverwerking zich aan hen zal manifesteren. Hierbij worden de aan de gebruikers bekende dataformaten (data-elementen, data-groepen, logische data-bases, schermen, berichten, etc.) en data-manipulaties (dialoogscenario's, logon-procedures, mogelijke vraagstellingen, etc.) vastgesteld.
- Keuze van de programmatuur en apparatuur nodig voor de implementatie van het gespecificeerde systeem. Deze keuze hangt uiteraard af van aanwezige kennis en mogelijkheden, financiële overwegingen, enz. Ook zullen in de praktijk vaak de specificaties hierover richtlijnen bevatten in de vorm van ontwerp-beperkingen.

Het technisch ontwerp stelt vast, hoe het te implementeren systeem aan de verlangde specificaties zal voldoen. M.a.w. het ontwerp hoe het systeem de verlangde informatieverwerking zal uitvoeren. Dit zal in het algemeen de structuur van data-bases, bestanden en data-processen opleveren. Hierbij vindt ook de verdeling plaats van de benodigde functies over aanwezige, gekochte of te maken systeem- en applicatie-software. Met name deze laatste aspecten zullen in het algemeen transparant zijn voor de gebruikers van het informatiesysteem.

De implementatie zal zich richten op het daadwerkelijke ontwerp van de data-declaraties en programmatuur, waarna de codering en het testen van de programmatuur volgt. Bovenstaande is slechts een korte schets van de hoofdgedachten. Het zij hier evenwel vermeld, dat een en ander zeker iteraties zal kennen, en dat onderverdelingen in subsystemen (bijv. naar aanleiding van specifieke user views) en toepassen van bijv. prototyping zal resulteren in fasering van de ontwikkeling per component van het systeem. Uitwerking van deze aspecten vallen echter buiten dit artikel.

1.3 Doelstellingen van INFOMOD

Om te komen tot een juist inzicht van de informatiebehoeften van een onderneming of organisatie (enterprise) is een model van de onderneming nodig. Dit model is echter gericht op de informatie-uitwisseling-aspecten. Het model dient het volgende te omvatten:

1. Een overzicht van de (relevante) bedrijfsfuncties en hun onderlinge samenhang.
2. Een overzicht van de dingen en zaken en hun onderlinge samenhang, waarover de bedrijfsfuncties informatie nodig hebben voor de vervulling van hun doelstelling.
3. Een overzicht van de informatiestromen hierover tussen de bedrijfsfuncties.
4. Een overzicht van de (inhoud van de) berichten die in deze stromen uitgewisseld worden.

Normaliter zal de analyse van de informatiebehoeften plaats vinden, omdat men voor de informatieverwerking en -uitwisseling computersystemen wil gebruiken. Hierbij is een op computertoepassing gebaseerd informatiesysteem een hulpmiddel voor deze informatie-uitwisseling en -verwerking. Het informatiesysteem is een medium via welke de informatie-uitwisseling tussen de bedrijfsactiviteiten plaatsvindt.

Het inzetten van een dergelijk informatiesysteem zal echter zijn invloed hebben op de inhoud en loop der informatiestromen. De daadwerkelijke informatie-uitwisseling immers zal schijnbaar plaatsvinden tussen een bedrijfsfunctie en het informatiesysteem. Dit zal in zekere zin een uitbreiding en aanpassing geven van de onderdelen 3 en 4 van het bedrijfsmodel.

In het (aangepaste) bedrijfsmodel is dit informatiesysteem geprojecteerd als een zwarte doos: We weten wel wat de doos voor ons doet, wat er ingaat, wat er uit zal of moet komen, maar niet hoe de doos inwendig is geconstrueerd, noch hoe hij werkt. Dit betekent ook, dat de specificaties voor een dergelijke doos zich hier aan dienen te houden. Op deze wijze krijgt enerzijds de constructeur van de doos de grootst mogelijke vrijheid in de keuze van de constructie en zal anderzijds de vaststelling van de informatie-behoefte van het bedrijf en de organisatie van de informatie-uitwisseling en -verwerking zo onafhankelijk mogelijk zijn van (toevallig) ad hoc voor handen zijnde computertechnieken en disciplines.

INFOMOD [6] is een methode en een taal voor de formulering van de onderdelen 2 en 4 van dergelijke bedrijfsmodellen. SADT [11] is een geschikte methode voor de onderdelen 1 en 3 en voor de algemene begeleiding van het proces om tot een bedrijfsmodel te komen, en wordt derhalve binnen Philips als zodanig te zamen met INFOMOD gebruikt.

De doelstellingen van INFOMOD kunnen dan ook als volgt geformuleerd worden:

- De specificatie van een model van de dingen, zaken en gebeurtenissen in hun onderlinge samenhang, **waarover** informatie in een informatiesysteem verwacht c.q. geëist wordt in termen van:
 1. Over welke dingen en zaken informatie nodig is:
Informatie Structuur Model,
 2. De onderlinge samenhang van deze dingen en zaken:
Informatie Regels,
 3. De berichtsamenstelling over deze dingen en zaken:
Informatie Operaties Specificaties (Berichten).
 4. De berichtuitwisseling met het informatiesysteem:
Informatie Operaties Specificaties (Acties).
- De specificatie dient - op het Informatieniveau volgens de Three Level Approach - inzicht in de

(conceptuele) betekenis van de voor het bedrijf (of bedrijfs onderdeel) relevante data te geven. Het Informatie Structuur Model en de Informatie Regels te samen definiëren de interpretatie van de door het informatiesysteem te verwerken en uit te wisselen informatie.

- De specificatie dient personen, die niet computer-technisch geschoold zijn, inzicht in bovenstaande aspecten te verschaffen. Hiertoe omvat INFOMOD grafische hulpmiddelen als entiteit-structuur en transitie-diagrammen (de zgn. 'ballen-platen'), etc.
- De specificatie zelf dient formeel, ondubbelzinnig en precies te zijn, zodat (in een later stadium) zij als input voor ontwerp- en programmeerhulpmiddelen kan dienen. De INFOMOD-taal voldoet volledig hieraan.
- De specificatie moet verschillende niveaus van detaillering en interesse toelaten. Hieraan beantwoordt het INFOMOD-submodelconcept.
- De specificatie dient op eenvoudige wijze aanpasbaar, wijzigbaar en uitbreidbaar te zijn. Het INFOMOD-submodelconcept is tevens hierop gericht.
- De specificatie moet onafhankelijk zijn van te kiezen computer-implementatie-technieken en apparatuur. De INFOMOD-informatiemodellen voldoen hieraan.

1.4 Uitgangspunten van INFOMOD

Bovenstaande doelstellingen hebben natuurlijk duidelijke consequenties. Daarom zijn de volgende uitgangspunten voor INFOMOD gekozen:

1. INFOMOD beschrijft de dingen en zaken waarover informatie nodig is en hun onderlinge samenhang. INFOMOD-modellen zijn 'object-georiënteerd'. Derhalve gaat de groepering en structurering van de beschrijvingen uit van de (ordering van de) dingen en zaken zelf. Dit houdt in, dat de groepering van bijv. de attributen bepaald wordt door de entiteiten en niet andersom. Entiteitdefinities worden niet bepaald door een mogelijke groepering van attributen, of, zoals bij sommige data-modeltechnieken, door een mogelijke groepering van informatie over de attributen. De representatievormen van de informatie, welke gebruiker-georiënteerd (extern - data) zijn, en die, welke gebruiker-transparant (intern) zijn, zijn aspecten van het data-niveau en het interne niveau volgens de principes van de Three Level Approach, en vallen derhalve buiten de sfeer van INFOMOD. (Zie hiervoor o.m. [2, 3, 4, 7].)
2. Het toepassen van INFOMOD zal in de specificatiefase geen enkele ontwerp- of implementatiebeslissing noodzakelijk maken. M.a.w. in de INFOMOD-taal of diagramtechnieken wordt geen onderscheid gemaakt tussen wat (straks) in een data-base zal zijn opgeslagen en wat door programmatuur afgeleid zal worden. Ook zal een informatie-analyst niet door INFOMOD gedwongen worden aan te geven wat door welk computer-component (bijv. applicatieprogramma) geïmplementeerd dient te worden. Dit houdt echter niet in, dat de INFOMOD-mogelijkheden dit in alle gevallen onmogelijk zou moeten maken. Integendeel, in INFOMOD kan men submodellen specificeren, die bijv. tot precies die zaken beperkt worden, waarover men de informatie in een data-base wenst op te slaan.
3. INFOMOD dient geen enkele restrictie op te leggen op het (gedeelte van) een bedrijf dat gemodelleerd dient te worden, noch op het gezichtspunt van een gebruiker op de gang van zaken in een bedrijf. Dit houdt in, dat INFOMOD een zeer rijk vocabulair moet bevatten. Dit betekent ook, dat, in het geval een bepaalde situatie desondanks niet (of niet gemakkelijk) te beschrijven is, INFOMOD aangepast dient te worden, c.q. de mogelijkheid moet hebben uitgebreid te worden. Met name dit uitgangspunt wijkt af van wat in de sfeer van data-processing gangbaar is. Toch wordt een dergelijk uitgangspunt steeds algemener beschouwd als noodzakelijk voor moderne software engineering.
4. Bovenstaand uitgangspunt betekent eveneens, dat een zeer sterk submodel-concept nodig is. Dit concept moet de modellering van allerlei invalshoeken en doorsneden mogelijk maken. De op deze wijze geformuleerde submodellen moeten op eenvoudige wijze geïntegreerd kunnen worden, c.q. geëxtraheerd uit het geïntegreerde model of andere submodellen. Nauw hiermee gelieerd is de definiëring van typen, supertypen en subtypen. Mede hierdoor staat in INFOMOD het classificatieprincipe centraal.
5. INFOMOD is gericht op **wat** er verwerkt moet worden, niet **hoe** het te verwerken. M.a.w. de INFOMOD-taal is in principe 'niet-procedureel'. De samenhang der dingen houdt echter ook tijdsafhankelijke samenhang in. De consequentie hiervan is, dat tijdsafhankelijkheden uiteraard beschrijfbaar zijn in INFOMOD. Dit leidt tot de mogelijkheid procedurele zaken te beschrijven. Dit zullen normaliter tijdsafhankelijke aspecten zijn van de dingen en zaken zelf, waarover informatie nodig is, en niet een beschrijving (programmering) hoe informatie hierover in het informatiesysteem te manipuleren.
6. De in INFOMOD geformuleerde specificaties zullen computer-implementatie-onafhankelijk zijn. Zij dienen echter formeel, precies en ondubbelzinnig te zijn. Dit houdt een formele en fundamentele basis in, die computer-onafhankelijk is. Deze noodzaak wordt tegenwoordig alom onderkend, met name voor 4e en 5e generatie computertoepassingen. INFOMOD is gebaseerd op formele logica en verzamelingenleer. Dit waarborgt mogelijkheden voor formele controle van de specificaties op juistheid en samenhang zonder contradicties.
7. De INFOMOD-taal moet echter computer-verwerkbaar zijn. Hierdoor is het mogelijk hulpmiddelen te ontwikkelen voor opstellen, aanpassen en controle van de specificaties en voor het ontwerpen van data-structuren en programmatuur ter implementatie van een informatiesysteem uitgaande van de specificaties. INFOMOD is echter onafhankelijk van de op dit ogenblik gangbare data- en programma-structuurtechnieken. (Ontwerp- en implementatie-hulpmiddelen zullen uiteraard wel afhankelijk hiervan zijn. Zij immers slaan de brug tussen de van de computertechnieken onafhankelijke specificaties en de van de computertechnieken afhankelijke implementaties.)

8. Hoewel de INFOMOD-taal een formele en voor computerverwerking geschikte taal moet zijn, dient zij toch dicht tegen de natuurlijke taal aan te liggen. Dit betekent, dat een betrekkelijke woordrijkdom aanwezig is, en dat ten behoeve van gebruikers van diverse pluimage dikwijls meer dan één manier van formuleren mogelijk moet zijn. Dit komt bijv. tot uiting in de mogelijkheid operatoren zowel in woorden, als ook in de gebruikelijke symbolen uit te drukken. Een ander gevolg is, dat INFOMOD versies kent voor verschillende nationaliteiten (o.a. een Engelse, Nederlandse, Franse en Portugese versie – een Duitse en een Italiaanse versie zijn in voorbereiding). Ook hier geldt, dat ondersteunende hulpmiddelen de verschillende mogelijkheden tot een enkele interne uitdrukkingvorm kunnen herleiden.

2 EEN CONTEXT VOOR INFORMATIESYSTEMEN

2.1 Informatiesystemen

Een van de elementaire eisen van de menselijke samenleving is informatie: Om te kunnen doen wat we willen of moeten doen hebben wij kennis nodig, en het is onmogelijk geworden voor ieder van ons om deze kennis uitsluitend met vallen en opstaan te vergaren door eigen ervaringen. Daarom moeten wij onze kennis ook verwerven door ons door anderen te laten informeren. Bijgevolg kan informatie gedefinieerd worden als: iedere soort kennis over dingen, feiten, concepten, enz. die uitwisselbaar is tussen gebruikers van deze kennis. De noodzaak informatie uit te wisselen is echter geenszins aan mensen voorbehouden. Met elkaar gekoppelde technische systemen, speciaal die in geautomatiseerde toepassingen, dienen ook informatie uit te wisselen, d.w.z. zij dienen elkaar te 'informeren' over de situatie en het gedrag van de zaken waarop of waarmee zij opereren. Ook hebben zulke systemen besturings- en vergelijkingsinformatie nodig.

Voor een uitwisseling van informatie is een verschaffer ('zender') van deze informatie nodig, en een gebruiker ('ontvanger'), die de informatie nodig heeft. Hierbij moeten twee belangrijke zaken worden opgelost:

- verschaffer en gebruiker bevinden zich (vaak) niet op dezelfde plaats – een verschil in afstand moet worden overbrugd (communicatiefunctie);
- verschaffer en gebruiker houden zich (vaak) niet op hetzelfde ogenblik bezig met de informatie – een verschil in tijd moet worden overbrugd (geheugenfunctie).

Een verschaffer van informatie en een gebruiker daarvan zouden beschouwd kunnen worden als twee delen van een 'systeem van ondernemen'. In een dergelijk systeem kan een derde component gedacht worden, die als een intermediair fungeert in de uitwisseling: een *informatiesysteem*. In deze context is een informatiesysteem een hulpmiddel om gebruikers behulpzaam te zijn in het ont houden en uitwisselen van informatie.

Om de uitwisseling van kennis – informatie – mogelijk te maken, moet deze noodzakelijkerwijs in een of andere overdraagbare vorm uitgedrukt, gerepresenteerd worden, bijvoorbeeld, geschreven of gesproken taal, beelden, data, enz. Informatiesysteem-ontwerp heeft zich in het algemeen vooral bezig gehouden met efficiënte me-

thoden en technieken voor omgaan met deze overdraagbare vormen (data). Hierbij werd het probleem deze vormen op de juiste wijze te interpreteren (de betekenis) vooral overgelaten aan de gebruiker (ontvanger) van deze vormen. Het doel van een uitwisseling van informatie is evenwel de uitwisseling van de kennis. Dit betekent dat een ontvanger de data moet interpreteren zoals de verschaffer deze bedoeld heeft. Daarom speelt de betekenis van datgene wat uitgewisseld wordt – de informatie – een zeer belangrijke rol. De representatievorm – de data – moet geenszins verwaarloosd worden. Het punt is echter, dat men alleen zinvol met de opslag en overdracht van de data kan omgaan, indien er een duidelijke overeenstemming en precieze afspraak is over wat de data voorstelt en welke interpretatieregels (semantiek) voor hen gelden.

Hier in Nederland wordt vaak onderscheid gemaakt tussen informatie, gegevens en data. Men gaat er dan van uit, dat gegevens pas informatie worden voor de ontvanger, indien hij er iets aan heeft. Hierbij komt dus het doel en de zinvolheid van de informatie ter sprake. In de uitgangspunten van het boven aangehaalde ISO-rapport, waarvan INFOMOD ook uitgaat, bepaalt dit concept van 'zinvolheid' en 'doel' of de ontvangen informatie kennisuitbreiding tot gevolg heeft. Dit is echter irrelevant voor het overdragen zelf: het informeren. Wil men dus een vergelijking maken tussen deze Nederlandse terminologie en de termen zoals in het ISO-rapport gehanteerd, dan kan de volgende lijst worden opgesteld:

informatie	↔	kennis (knowledge)
gegevens	↔	informatie (information)
data	↔	data

In dit artikel zullen wij, conform ISO, de tweede reeks termen hanteren.

2.2 De Omgeving (van een informatiesysteem)

Het doel van een informatiesysteem is de verschillende delen van een systeem van ondernemen bij te staan in de uitwisseling van informatie. Deze componenten, die het informatiesysteem gebruiken, vormen de *omgeving* van het informatiesysteem. Onder een gebruiker wordt verstaan: diegene of datgene wat gebruik maakt van het informatiesysteem. Dit kunnen mensen zijn, of machines, andere informatiesystemen, enz. Een gebruiker communiceert met het informatiesysteem door middel van berichten. Vanuit het informatiesysteem bekeken is de omgeving de bron en de bestemming van de berichten.

Een informatiesysteem opereert alleen volgens vanuit de omgeving verstrekte instructies. In deze kan het zelf geen werkelijke initiatieven nemen, hoewel dit soms schijnbaar wel zo kan zijn. In dit geval is dit echter naar aanleiding van eerder ontvangen instructies. Toch kunnen gebruikers wel degelijk beperkt worden in hun mogelijkheden het systeem te gebruiken. Bijvoorbeeld niet iedereen mag informatie in het systeem inbrengen. Anderen kunnen beperkt worden in welke informatie opgevraagd mag worden. Weer anderen, mogelijk zelfs zeer weinigen, mogen instructies verstrekken om het systeem zelf te wijzigen.

2.3 Het Universum van Discussie

De doelstelling van een informatiesysteem is het de gebruikers mogelijk te maken feiten uit te wisselen over dingen en zaken. De verzameling dingen en zaken waar-

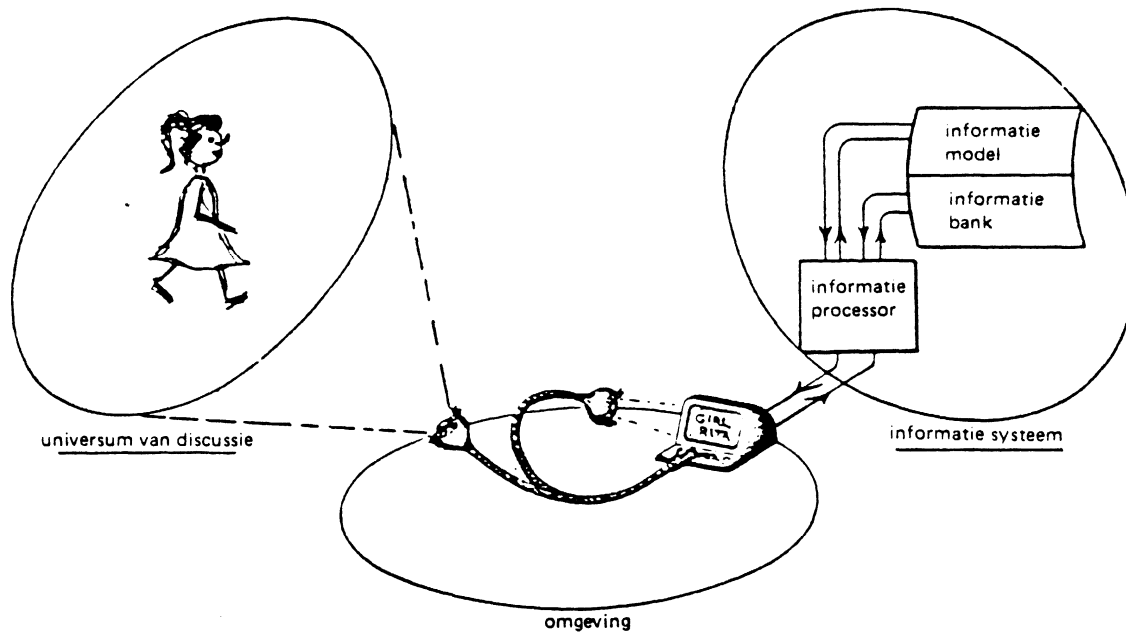
over deze communicatie plaatsvindt, deze discussie gevoerd wordt, wordt het *universum van discussie* genoemd. Dit universum van discussie is een gedeelte van een reële of veronderstelde wereld.

Alle dingen en zaken, die beschouwd worden tot het universum van discussie te horen, worden entiteiten genoemd. Zij kunnen concreet zijn, zoals bijv. een televisieapparaat of Jan. Zij kunnen echter ook abstracte dingen en zaken zijn, zoals bijv. een aankoop of de organisatiestructuur van een bedrijf. Men kan ook soorten en klassen van entiteiten onderkennen, bijv. personen, afdelingen. Een dergelijke classificatie is gebaseerd op overeenkomsten tussen deze entiteiten, rekening houdend met eigenschappen van de entiteiten en situaties waarin zij kunnen verkeren. De keuze van deze overeenkomsten is eigenlijk arbitrair en berust meestal op pragmatische gronden, gebaseerd op de redenen waarom men een bepaald universum van discussie wil beschouwen.

Informatie over deze entiteiten beschrijft hun eigenschappen, omstandigheden waarin zij verkeren en gebeurtenissen waarin zij betrokken zijn. In het algemeen is alles wat tot een universum van discussie gerekend wordt afhankelijk van de tijd. D.w.z. dit alles kan in de loop van de tijd veranderen. Dit is niet alleen waar voor de individuele entiteiten en hun omstandigheden, maar dit geldt ook voor de regels, wetten, indeling naar soorten, etc. Meestal verloopt het veranderingsproces hiervoor echter langzamer.

2.4 Het Ondernemingssysteem

Het *ondernemingssysteem* bestaat uit het universum van discussie, de Omgeving en het Informatiesysteem, als afgebeeld in figuur 2.1.



Figuur 2.1: Informatiesysteem, Omgeving en Universum van Discussie

Het universum van discussie omvat alle dingen, zaken en gebeurtenissen, *waarover* informatie vereist wordt; de Omgeving wordt gevormd door alle gebruikers *waarvoor* de informatie vereist is; het informatiesysteem behelst alle mechanismen *waarmee* de informatie bewaard en

uitgewisseld wordt door de gebruikers. Deze onderverdelingen van het ondernemingssysteem kunnen echter geheel of gedeeltelijk over elkaar heen vallen. D.w.z. het is zeer wel mogelijk, dat een persoon of een ondernemingsactiviteit informatie nodig heeft – dus tot de Omgeving gerekend wordt – en tegelijkertijd ook onderwerp is waarover informatie nodig is – dus tot het universum van discussie behoort. Ook zal men vaak informatie over het informatiesysteem zelf willen hebben. In dat geval is het informatiesysteem zelf onderdeel van het universum van discussie.

3 HET INFORMATIEMODEL

3.1 De rol van een Informatie Model

Communicatie is alleen mogelijk indien er een gemeenschappelijke overeenstemming is over de dingen en zaken, die deel uit maken van het universum van discussie. Anderzijds zijn communicatiestoornissen dikwijls terug te voeren op het ontbreken van een gemeenschappelijke afstemming van het bedoelde universum van discussie. Uitwisseling van informatie moet bij voorkeur precies en ondubbelzinnig zijn, zeker wanneer de communicatie met behulp van computers plaatsvindt. Daarom dient een model van het universum van discussie geconstrueerd te worden, te meer omdat computers hooguit een model (beschrijving) van de werkelijkheid kunnen interpreteren, en niet de werkelijkheid zelf.

Dit model van het universum van discussie, dat als doel heeft vast te stellen over welke dingen, zaken, omstandigheden en gebeurtenissen informatie nodig is, wordt *Informatiemodel* (of *conceptueel schema*) genoemd. Dit informatiemodel beschrijft welke entiteiten mogelijk in het universum van discussie kunnen voorkomen, d.w.z.

welke er ooit bestaan hebben, nu bestaan kunnen of in de toekomst ooit bestaan zouden kunnen. Het beschrijft op dezelfde wijze welke feiten, omstandigheden en gebeurtenissen mogelijk zijn. Ook kunnen algemenere eigenschappen die voor verscheidene entiteiten gelden, of hen

onderscheiden van andere groepen, beschouwd worden. Zo kunnen ook regels en wetten onderkend worden volgens welke entiteiten zich kunnen of moeten gedragen (bijv. een medewerker kan tot niet meer dan één afdeling tegelijk behoren). Dergelijke regels en wetten hebben betrekking op het gedrag of de omstandigheden van entiteiten in het universum van discussie. Juist deze algemenere wetten, classificaties, etc. zal een informatiemanalysist in het informatiemodel willen vastleggen in overleg met degenen waarvoor het informatiemodel en het informatiesysteem bedoeld is. Deze algemenere regels vormen als het ware een blauwdruk van het gedeelte van een wereld waarover zij geïnformeerd willen worden.

Omdat dit informatiemodel een model is van de dingen, zaken, enz. waarover de informatie handelt, kan het tevens dienen als een model voor de inhoudelijke betekenis van de informatie, het is het conceptuele model voor het informatiesysteem. Merk op, dat het alleen dan een model van het informatiesysteem (of delen daarvan) is, indien dit informatiesysteem zelf onderwerp van informatie is, dus deel is van een universum van discussie. Een informatiemodel behelst een gemeenschappelijke overeenkomst hoe de dingen en zaken in het universum van discussie te beschouwen en wat geldt voor hun gedrag en omstandigheden, voor zover dit relevant geacht wordt. Daarom ligt het hiermee ook vast wat voor informatie in het informatiesysteem kan voorkomen en welke veranderingen hierin kunnen optreden.

Uitwisseling van bedoeling (semantiek van de communicatie) is het essentiële doel van het gebruik van een informatiesysteem. Evenals bij gebruik van natuurlijke taal, kan men weliswaar horen of lezen wat gezegd of geschreven wordt zonder het te begrijpen, maar communicatie houdt overeenstemming en begrijpen van wat bedoeld wordt in. Een informatiemodel specificeert derhalve de betekenis van de informatie in het informatiesysteem en is de feitelijke overeenkomst tussen de gebruikers van het informatiesysteem over de vaststelling van het universum van discussie. Deze overeenkomst kan in de loop der tijd veranderd worden, maar op ieder moment legt het vast welke informatie in het informatiesysteem al dan niet aangetroffen kan worden.

Zoals reeds eerder opgemerkt, kan informatie alleen uitgewisseld worden, indien het in een of andere vorm (data) uitgedrukt is. Deze vormen kunnen op hun beurt ook weer aan verandering onderhevig zijn, bijv. omdat sommige gebruikers andere vormen prefereren, of omdat de computertechnologie verandert. Daarom is het belangrijk een representatie-onafhankelijke specificatie van de betekenis van deze vormen te hebben. Dit wordt door het informatiemodel bereikt. Vele verschillende representatie-vormen zijn denkbaar en zinvol. Een gebruiker moet zo vrij mogelijk gelaten worden in zijn keuze van de aan de buitenkant van het informatiesysteem zichtbare vormen. Evenzo moet de ontwerper van een informatiesysteem in staat zijn voor de interne representatie de meest efficiënte technieken toe te passen. De betekenis van de informatie mag hier echter niet door veranderen. Het informatiemodel vormt dan ook de specificatie van de semantiek van al deze vormen.

Zo kunnen de volgende doelstellingen voor een informatiemodel geformuleerd worden:

1. Het legt een gemeenschappelijke basis vast voor

het beschouwen van het gedrag van het universum van discussie.

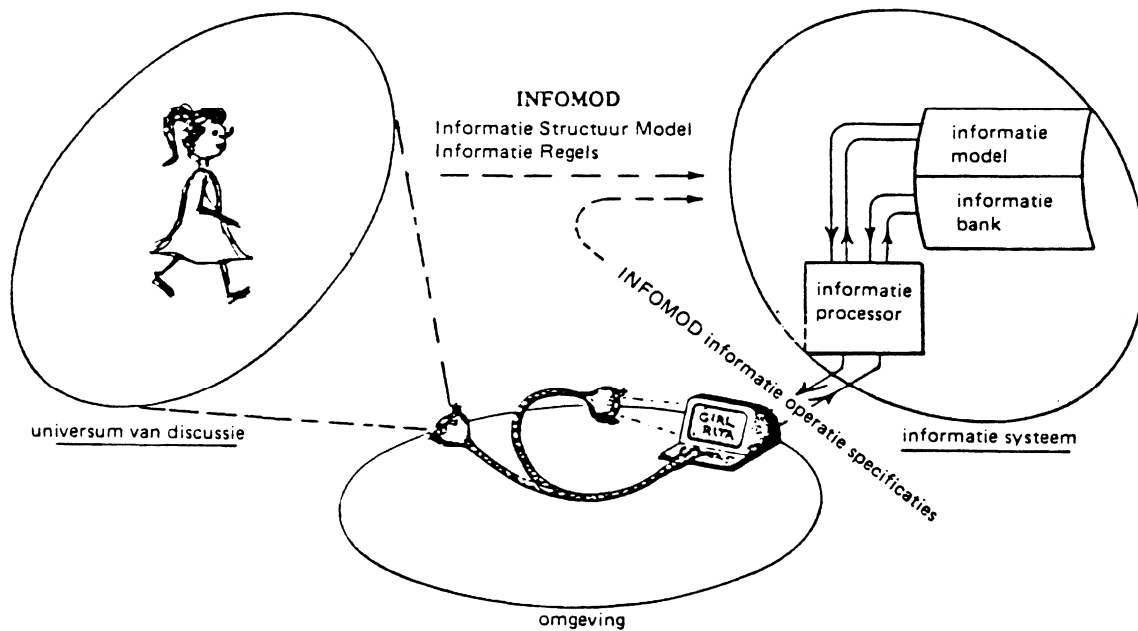
2. Het definieert de toegestane veranderingen en verwerking van de informatie over het universum van discussie in het informatiesysteem.
3. Het vormt een basis voor de interpretatie van de externe en interne vormen (data) waarin de informatie over het universum van discussie gerepresenteerd wordt.
4. Het vormt de semantische specificatie voor de implementatie van het informatiesysteem.

3.2 De drie delen van een Informatiemodel

Een andere kijk op een informatiemodel ontstaat indien nagegaan wordt welke mogelijke aspecten voor het informatiesysteem gespecificeerd worden. In grote lijnen worden drie soorten zaken beschreven, de entiteiten en hun eigenschappen en omstandigheden, de gedragsregels van het universum van discussie, en de verlangde informatie-uitwisseling hierover tussen informatiesysteem en omgeving. Zo worden drie delen onderkent: het informatie-structuur-model, de informatieregels en de informatie-operaties-specificaties.

De specificatie welke entiteiten in het universum van discussie kunnen voorkomen en in welke situatie zij een rol kunnen spelen, wordt het *informatie-structuur-model* genoemd. INFOMOD kent een eenvoudige maar sterke diagramtechniek om deze structuur van onderling samenhangende entiteiten uit te beelden. Deze structuur vormt tevens de basis van de structuur van de informatie hierover. Dergelijke diagrammen dienen echter alleen om een overzicht te verschaffen. De exacte definitie van de structuur en zijn elementen wordt beschreven in de formele INFOMOD-taal. Bij de implementatie van een informatiesysteem vormt dit informatiestructuur-model de grondslag voor met name het ontwerp van een database of de bestanden.

De *informatieregels* specificeren de regels en condities van de omstandigheden waarin de entiteiten in het universum van discussie betrokken kunnen zijn. Dit beschrijft dus wat met de entiteiten kan gebeuren. Ruwweg zijn zij onder te verdelen in tijdsafhankelijke en tijdsafhankelijke regels. Een tijdsafhankelijke regel definieert wat voor een entiteit geldt los van de tijd, bijv. 'Iedere medewerker heeft één en niet meer dan één salaris-nummer'. Omdat dit meestal min of meer onveranderlijke situaties behelst, worden zij ook wel 'statische' regels genoemd. De tijdsafhankelijke regels beschrijven meestal mogelijke veranderingen (transitieregels), of zij houden mogelijke veranderingen in (tijdstempels). Een voorbeeld van een transitieregel is: 'De burgerlijke stand van een ongehuwd persoon kan alleen maar veranderen in gehuwd'. Soms worden deze regels 'dynamische' regels genoemd. De INFOMOD-taal maakt de formulering van beide soorten regels mogelijk op een manier, die dicht tegen de natuurlijke taal aanligt. In het algemeen kunnen regels (behalve de zeer eenvoudige) niet goed in diagrammen worden uitgedrukt. Voor overzichten van tijdsvolgorde van situaties en mogelijke overgangen van situaties in elkaar biedt INFOMOD transitiedia-gram-technieken, die aansluiten op die voor de structuur. Deze regels over het gedrag in het universum van discussie kunnen gebruikt worden om de integriteit van de informatie in het informatiesysteem te bewaken. Integriteit betekent dat er geen tegenspraken voorkomen in



Figuur 3.1: De drie delen van een Informatiemodel

de informatie. Afdwinging van deze regels in het informatiesysteem wordt in het algemeen bewerkstelligd door controle- en afleidingsroutines, die zowel deel kunnen uitmaken van de systeemsoftware als de applicatiesoftware.

De *informatie-operaties-specificaties* specificeren de verlangde informatie-uitwisseling tussen het informatiesysteem en de gebruikers in de omgeving. Zij beschrijven de inhoud van de berichten, die bij deze uitwisseling gebruikt worden, en de condities voor hen. Hieronder vallen bijvoorbeeld de formulering van de selectie-criteria en vragen voor informatie te verkrijgen uit het informatiesysteem. Ook kunnen deze condities tijdsvolgorden aanduiden, waarin de uitwisselingen (de acties) dienen plaats te vinden. Op deze manier vormen de informatie-operaties-specificaties de definiëring van de semantische aspecten van het interface tussen gebruikers en informatiesysteem. De precieze definiëring wordt in de INFOMOD-taal uitgedrukt. Hoewel deze specificaties in het algemeen in een definitionele vorm worden beschreven, laat de INFOMOD-taal ook 'procedurele', d.w.z. in tijdsvolgorde uitgedrukte beschrijvingen toe. Hiernaast biedt INFOMOD diagram-technieken om een overzicht te geven tussen de tijdsvolgordelijke samenhang tussen bericht-uitwisselingen. De informatie-operaties-specificaties specificeren op deze manier, welke informatie de applicatieprocedures op welke momenten onder welke condities aan het informatiesysteem moeten toevoegen, c.q. vanuit het informatiesysteem moeten verstrekken. Zij beschrijven niet hoe de informatie binnen het informatiesysteem verwerkt wordt. Dit is het onderwerp van het logisch en technisch ontwerp van het informatiesysteem. Ook beschrijven zij niet in welke vorm de informatie-uitwisseling – de opmaak van de berichten – plaats moet vinden. Dit laatste is het onderwerp van de data-modellering op het data-niveau.

3.3 Het Submodel Concept

Historisch gezien zijn er twee manieren om informatiesystemen te ontwerpen:

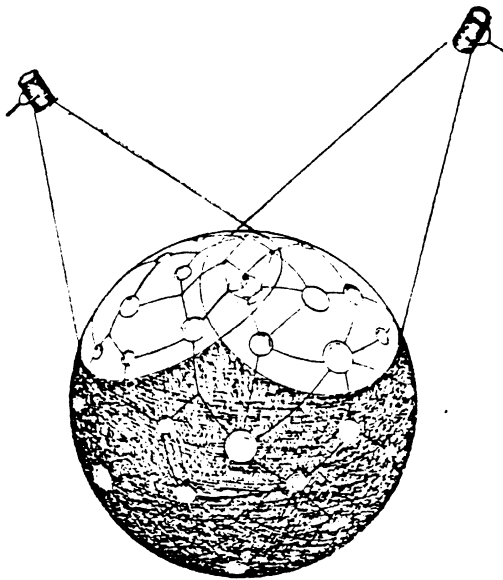
1. Ontwerp ieder systeem onafhankelijk van ieder ander.
2. Ontwerp een geïntegreerd systeem dat alles omvat.

De eerste benadering ondersteunt historisch organisatorisch gedrag en legt de nadruk op lokale optimalisatie. Het resultaat is systemen die moeilijk aan te passen en te gebruiken zijn voor andere dan hun oorspronkelijke doeleinden. Overkoepelende systemen, bijvoorbeeld voor het verzamelen van bestuurlijke informatie voor directies, worden bijzonder ingewikkeld, omdat zij allerlei conflicten in gerichtheid en uitvoering tussen de verschillende systemen moeten oplossen. Ook is het bijzonder moeilijk zo niet onmogelijk regels te handhaven over deze verschillende systemen heen, zodat informatie verkregen vanuit verschillende systemen vaak vol tegenstrijdigheden is.

De tweede benadering ondervindt ondermeer de volgende problemen:

- het systeem is zo groot en complex dat niemand alle facetten kent. Derhalve is het moeilijk het geheel volledig te begrijpen.
- In het belang van de goede verstandhouding bevat het ontwerp zo veel compromissen dat een gebruiker het antwoord op zijn individuele probleem nauwelijks of niet kan herkennen.
- Het duurt zo lang voordat het systeem ontworpen en geconstrueerd is, dat de oorspronkelijke probleemstelling al veranderd is voor het systeem is voltooid.

Willen informatiesystemen efficiënte hulpmiddelen zijn voor de gebruikers ervan, dan moeten zij meer in overeenstemming zijn met de manier waarop de gebruikers hun wereld, hun universum van discussie, waarover zij informatie nodig hebben, beleven. Daarnaast moeten de systemen ook beter tegemoet komen aan de manier waarop de gebruikers deze informatie kunnen en willen verstrekken c.q. ontvangen.



Figuur 3.2: De Metaphor van de Zoeklichten

De Metaphor van de Zoeklichten – iedere gebruiker of gebruikersgroep (ondernemingsactiviteit) heeft zijn eigen zoeklicht op dat deel van de wereld, waarover hij informatie nodig heeft om te kunnen doen wat hij wil of moet doen – is een van de grondstellingen waarvan INFOMOD uitgaat. Het submodel-concept is dan ook fundamenteel in INFOMOD. Dit gaat uit van de volgende doelstellingen:

1. Iedere gebruiker of gebruikersgroep moet zijn eigen submodel kunnen construeren volgens het beeld dat hij heeft van het deel van een wereld, waarover hij informatie moet of wil hebben. Dit gedeelte is *zijn* universum van discussie.
2. Aansluiting en overeenstemming met andere submodellen is alleen noodzakelijk voor die delen van de submodellen, die een gemeenschappelijk (deel van een) universum van discussie beschrijven. Dit behelst ook de regels die gelden voor dit gemeenschappelijke gedeelte.
3. Een gebruiker moet niet (door het informatiesysteem) gedwongen worden de visie of een submodel van een ander te accepteren als zijn eigen submodel. Het moet mogelijk zijn verschillende namen te kiezen voor dezelfde dingen, en om verschillende indelingen naar soorten te hebben.
4. De samenvoeging van de diverse submodellen resulteert in het geïntegreerde informatiemodel voor een informatiesysteem. Vanzelfsprekend zullen hierbij aansluitingsproblemen moeten worden opgelost. Het resultaat moet naderhand uitbreidbaar zijn met weer andere submodellen, mits deze niet in tegenspraak zijn met elkaar of met het reeds aanwezige geïntegreerde model.
5. Een gebruiker moet ook in het geïntegreerde systeem zijn submodel kunnen terugvinden. Een gebruiker moet zoveel als maar enigszins mogelijk het informatiesysteem kunnen gebruiken binnen de sfeer van zijn submodel zonder dat hij enige kennis hoeft te hebben over niet gemeenschappelijke delen van andermans submodellen. Het is duidelijk, dat hij niet al het andere kan negeren. Er kunnen

immers algemene regels gelden voor een wereld waarbinnen zijn interessesfeer valt, ook al heeft hij deze regels niet onderkent of voor hem niet belangrijk geacht. Nochtans gelden deze regels wel, of hij wil of niet. Toch moet hij in staat gesteld worden die zaken te negeren, die voor hem irrelevant zijn, ook al dient het informatiesysteem daar wel rekening mee te houden.

4 EEN RAAMWERK VOOR EEN INFORMATIESYSTEEM

4.1 De Drie Niveaus Benadering

Hoewel iedere uitwisseling van informatie noodzakelijkerwijs in een of andere vorm moet plaatsvinden, komt toch de interpretatie van deze vorm op de eerste plaats. De representatievorm wordt van secundair belang geacht, hoewel zeker niet irrelevant. Feitelijk kunnen drie belangrijke groepen aspecten onderkend worden bij informatie-uitwisseling via informatiesystemen:

1. Wat is er nodig, c.q. wat wil men weten.
2. Hoe ziet dit er aan de buitenkant van het informatiesysteem voor de gebruiker uit.
3. Hoe wordt dit binnenin het informatiesysteem (computersysteem) verwezenlijkt.

De eerste groep is gericht op welke informatie over welke dingen en zaken in het universum van discussie onthouden of verwerkt moet worden in het informatiesysteem. Dit zijn de informatie (of conceptuele) aspecten van een informatiesysteem. De tweede groep concentreert zich op hoe data de informatie representeert, hoe deze data aan het informatiesysteem aangeboden kan worden, en hoe zij opgevraagd kan worden. Dit zijn de data (of externe) aspecten van een informatiesysteem. Merk op, dat deze zowel de data-definities als de data-manipulaties omvatten, zoals deze door de gebruiker aan de buitenkant van het informatiesysteem ervaren worden.

Beide bovenstaande groepen zijn het belangrijkste voor de gebruikers. Zij zijn direct betrokken bij de keuzebepalingen hiervoor en krijgen tijdens gebruik met deze aspecten te maken.

Voor de ontwerpers en constructeurs van een informatiesysteem is ook de derde groep aspecten belangrijk. Deze behelst onderwerpen als hoe de data intern in het (computer)systeem op te slaan en te verwerken en hoe de betrouwbaarheid en integriteit van de informatie gewaarborgd wordt. Dit zijn de interne aspecten van een informatiesysteem. Deze zaken zijn (normaliter) niet zichtbaar voor de gebruikers, zij zijn transparant voor hem.

Historisch gezien, richtte het informatiesysteem-ontwerp zich voornamelijk op de data-aspecten en interne aspecten. Men begon met een inventarisatie en analyse van de data en het gebruik hiervan. Er werden data-modellen geconstrueerd, die de samenhang tussen de data-elementen en hun verwerking beschreven. Deze methode is in bepaalde omstandigheden heel wel uitvoerbaar, maar gaat van twee wezenlijke vooronderstellingen uit:

1. Het gedrag van het universum van discussie kan afgeleid worden van het gedrag van (het gebruik van) de data.
2. De informatiebehoefte van de huidige en toekomstige gebruikers kan afgeleid worden uit de huidige en nu gewenste data.

Beide vooronderstellingen zijn op zijn minst aanvechtbaar. De aanpak was echter redelijk, gegeven de voorhanden zijnde ontwerp- en implementatietechnieken van de zeventiger jaren. Het zou onjuist zijn te stellen, dat een aanpak waarbij doelstelling en ontwerp van een informatiesysteem gevonden wordt door abstrahering vanuit de aanwezige data en huidige data-verwerking, nutteloos zou zijn.

Desondanks is het niet erg waarschijnlijk dat een dergelijke aanpak een systeem zal opleveren, dat gedurende lange tijd echt bruikbaar blijft. Aanwezige data is altijd een afspiegeling van op dit moment gangbaar gedrag van een organisatie, en dit is juist aan regelmatige verandering onderhevig. Zelfs als data abstractiemethoden de entiteiten waarover het gaat correct vaststellen, dan nog worden de regels, die voor hun gedrag gelden, niet gevonden, omdat zij niet gereflecteerd worden in de data zelf – zij zijn niet vastgelegd in de representatie vormen. Juist de meeste moderne data-abstrahiemethoden en technieken (waaronder de diverse normaalvormen van de relationele data-modellen) hebben letterlijk geen enkele manier om, gebruikmakend van de methode zelf, die regels uit te drukken, die gevonden worden als men bij de analyse uitgaat van de dingen zelf en hun omstandigheden, in plaats van uit te gaan van de representatie vormen van informatie daarover.

Tegenwoordig wordt meer en meer onderkend, dat de informatieaspecten de meest belangrijke zijn. Dat wil zeggen, dat eerst vastgesteld moet worden waarover informatie door het informatiesysteem verwerkt moet worden, en wat voor deze dingen en zaken geldt. Daarna pas wordt vastgesteld hoe deze informatie te representeren en hoe deze te hanteren in de vorm van data. Als laatste kan dan een zinvol en efficiënt systeem geconstrueerd worden om de gebruikers als hulpmiddel te dienen voor het bewaren en verwerken van de data.

Het onderscheid tussen de betekenis van de informatie en de verschillende representatievormen is duidelijk beschreven in wat bekend geworden is als de Drie Niveaus Benadering (Three Level Approach) [1, 5]. De drie componenten hiervan zijn:

- het informatie (of conceptueel) niveau;
- het data (of externe) niveau;
- het interne niveau.

Alle aspecten, die te maken hebben met de betekenis en interpretatie van informatie worden geacht zich op het informatieniveau te bevinden. Dit omvat zowel de specificatie, als ook de manipulatie en beheersing van de informatie in termen van betekenis. De talen voor informatiemodellering zijn gericht op beschrijvingen op dit niveau.

Het data-niveau omvat alle aspecten, die te maken hebben met hoe de gebruiker de informatie en de verwerking hiervan ziet en ervaart. Dit is de informatie zoals deze zich manifesteert aan de buitenkant (het externe interface) van het informatiesysteem. Dit wordt ook wel het gebruikers-interface tussen het informatiesysteem en zijn omgeving genoemd. Dit behelst zowel de representatie van de informatie – de data – op een voor de gebruiker gemakkelijke manier, als ook de manipulatie van deze data door de computer, voor zover noodzakelijkerwijs zichtbaar voor de gebruiker.

Op het interne niveau worden alle aspecten in beschouwing genomen, die te maken hebben met de inwendige uitvoering en verwerking van de informatie binnen het

(computer) systeem. Deze aspecten zijn normaliter transparant voor de gebruiker, d.w.z. de gebruiker heeft hiervan geen weet te hebben. Hieronder vallen zulke onderwerpen als:

- de interne (fysieke) opslagvormen van de informatie in het geheugen en op discs, etc.;
- technieken en maatregelen om de interne efficiëntie van de computerprocessen te bevorderen, en keuze en uitvoering van benaderingsmechanismen en technieken, zoals tabelvormen, sorteringen, etc.;
- regulering van gelijktijdig gebruik, herstelprocedures, gedistribueerde opslag, etc.;

4.2 De Drie Niveaus Architectuur.

Gebaseerd op de Drie Niveaus Benadering kan een Drie Niveaus Architectuur [1, 5] als volgt gezien worden:

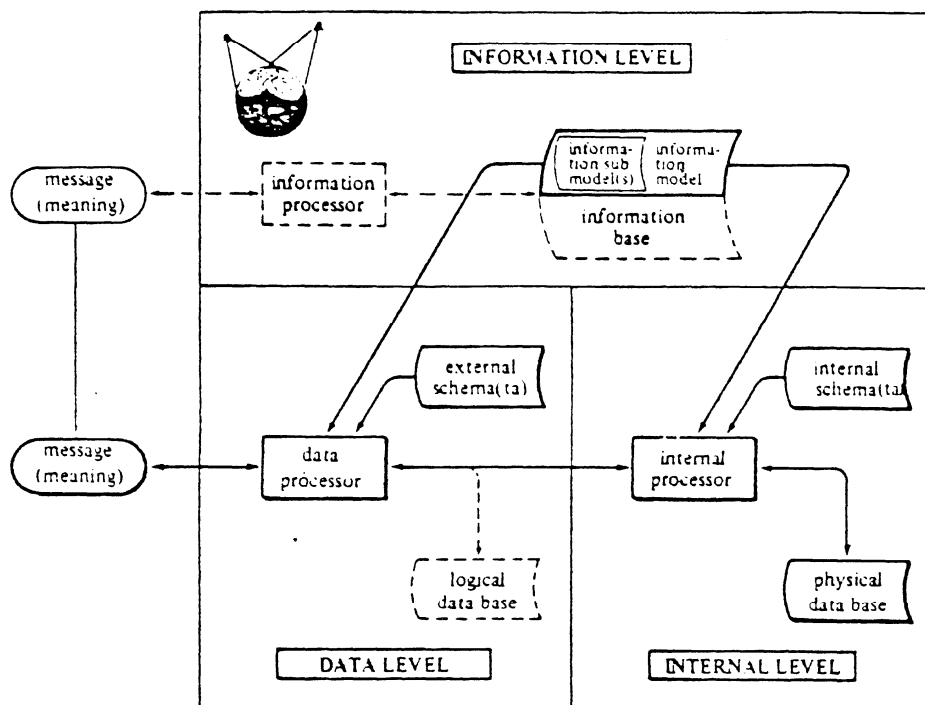
Op het informatieniveau neemt men aan dat alle informatie, die aan het informatiesysteem is toevertrouwd, c.q. daaruit opvraagbaar is, te zamen een informatiebank (information base) vormt. Aangenomen wordt, dat alle verwerking, controle en regulering van deze informatie uitgevoerd wordt door een informatie-verwerkingsmechanisme (information processor). Feitelijk wordt dit mechanisme gevormd door alle hardware, software (en eventueel operateurs), die de informatieverwerking in het systeem uitvoeren op last van en ten behoeve van de gebruikers in de omgeving. In een geautomatiseerd informatiesysteem bestaat dit mechanisme uit de computer zelf met de systeem software, database management software, en de applicatieprogramma's. De informatiebank wordt geïmplementeerd door de inhoud van de fysieke database te zamen met de programmatuur, die hieruit allerlei additionele informatie afleiden en genereren. De taak van het mechanisme van de informatieregels te handhaven wordt in werkelijkheid gedeeltelijk uitgevoerd door controleroutines van een database managementsysteem en andere systeemsoftware, gedeeltelijk door applicatieprogrammatuur.

De informatieverwerking wordt op het informatieniveau als volgt gezien:

Een bericht, dat bijv. nieuwe informatie bevat, wordt aan het informatiesysteem aangeboden. Het verwerkingsmechanisme zal, gestuurd door de in het informatiemodel vastgelegde regels en mogelijk anderszins onder raadpleging van reeds in het systeem aanwezige informatie, de nieuwe informatie opnemen in het systeem, of deze weigeren omdat zij in tegenspraak is met al aanwezige informatie of de regels. Opvraging van informatie geschiedt door een daartoe strekkende vraag aan het mechanisme aan te bieden. Hierop zal het mechanisme de aanwezige informatie raadplegen en een passend antwoord geven in de vorm van een bericht naar buiten.

Zoals reeds bij de bespreking van het submodel-concept is vermeld, is een gebruiker in de praktijk alleen in zijn eigen universum van discussie geïnteresseerd. Deze gebruikersvisie wordt vastgelegd in een informatiesubmodel (conceptueel subschema). Alle informatiesubmodellen samen vormen het informatiemodel (conceptueel schema), dat geldt voor het informatiesysteem.

Op het data-niveau spelen de door de gebruikers gekozen data-vormen een rol. Hier zijn de opmaak en uitvoeringsvormen van de berichten (schermen, rapporten, etc.) gedefinieerd. Hierbij heeft de gebruiker ook zeker een be-



Figuur 4.1: Drie Niveaus Architectuur van een Informatiesysteem

paalde voorstelling hoe de voor hem relevante informatie in het systeem aanwezig is. Dit is zijn logische database. Deze data-definities worden gespecificeerd in de zogenaamde externe schemata. Deze externe schemata definiëren de representatievormen van de informatie die voor een gebruiker in zijn informatiesubmodel is gespecificeerd. Zij zijn dus ook in die zin gebruiker-georiënteerd. De werkelijke uitwisseling van de berichten tussen systeem en gebruiker vindt plaats op het data-niveau. De mechanismen die dit uitvoeren zijn applicatieprocessen, query-processoren, rapport-generatoren, etc. Zij vormen zo te zeggen de communicatie-mechanismen met de gebruiker.

Deze logische database van een gebruiker is echter niet werkelijk bestaand. Het is slechts een 'afbeelding' van wat er voor hem relevant is en in het systeem aanwezig verondersteld wordt. Dit betekent, dat het informatiesysteem ook zulke taken moet verrichten als:

- de integratie van de activiteiten ten behoeve van de verschillende gebruikers,
- de transformatie van de voor de individuele gebruikers bestemde representatie vormen.

Hiertoe kunnen de logische databases van verschillende gebruikers samengevoegd worden tot een database op groepsniveau, enz. tot uiteindelijk een totaal beeld van een logische database voor alle gebruikers ontstaat. Deze laatste database wordt in het database schema gedefinieerd. Merk op, dat het hier om data, d.w.z. representatie van informatie handelt. De semantiek van deze data ligt in het informatiemodel vast. In de hedendaagse database techniek wordt deze logische database op het data-niveau gedacht. Feitelijk hoeft echter geen een gebruiker deze complete database te kennen. Zij kan dus transparant zijn voor hen. Dit betekent, dat deze gemeenschappelijke database in dat geval op het interne niveau

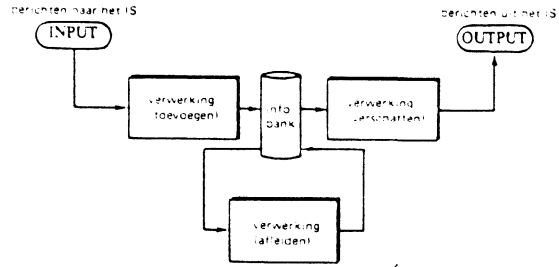
thuishoort, en dat dus dit database schema een intern schema zou moeten zijn.

Binnenin het informatiesysteem is een groot gedeelte van de informatie feitelijk opgeslagen in fysieke databases (of bestanden). Andere informatie wordt hieruit afgeleid door afleidingsprocessen. Deze te zamen zijn de werkelijke implementatie van de logische database zoals de gebruiker zich die voorstelt. De fysieke databases worden in zogenaamde interne schemata gedefinieerd. Vaak is er sprake van één fysieke database in het informatiesysteem, maar dit hoeft geenszins altijd het geval te zijn. In die gevallen waar het informatiesysteem de communicatie tussen geografisch verspreide gebruikers ondersteunt, is het zeer wel mogelijk, dat delen van de informatie fysiek lokaal worden opgeslagen. In zulke gevallen is er sprake van gedistribueerde databases. Bij voorkeur is deze fysieke spreiding transparant voor de gebruiker en derhalve een aspect op het interne niveau. Aangezien INFOMOD, het onderwerp van dit artikel, zich op het informatieniveau richt, valt een nadere bespreking van ontwerp en implementatieaspecten van het data en interne niveau buiten de doelstellingen van dit artikel, hoe belangrijk deze niveaus ook mogen zijn [2, 3, 4, 7].

4.3 Principes voor Informatie-uitwisseling

In principe zijn er twee aanleidingen voor een informatie-uitwisseling tussen een gebruiker in de omgeving en het informatiesysteem. De eerste aanleiding is een gebeurtenis, die plaatsvindt of plaatsgevonden heeft in het universum van discussie en waarover de gebruiker informatie rapporteert aan het informatiesysteem - er wordt nieuwe informatie toegevoegd. De tweede is, dat een gebruiker informatie wenst over iets in het universum van discussie - er dient informatie aan de gebruiker gerap-

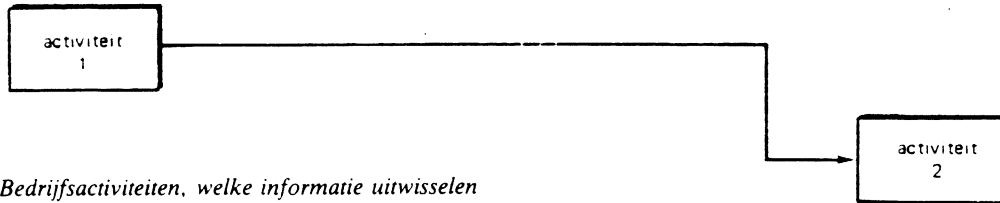
porteed te worden door het informatiesysteem. In de praktijk kunnen veel ingewikkelder informatie-uitwisselingen plaatsvinden, maar deze zijn in het algemeen altijd te herleiden tot combinaties van deze twee basismogelijkheden. INFOMOD gaat er van uit, dat informatie altijd gaat over iets in het universum van discussie. Dit kan gaan over een situatie – dingen en zaken en hun onderlinge samenhang – of over een gebeurtenis – in feite een verandering van een situatie. Hierbij doet het er in principe niet toe of de situatie of gebeurtenis een gevolg is van een activiteit van een gebruiker in de omgeving. Dit is zeer goed mogelijk en in dat geval vindt de gebeurtenis plaats in een stukje van de 'wereld' dat zowel deel uitmaakt van het universum van discussie als van de omgeving van het informatiesysteem. Voor de rapportage hiervan is dit echter irrelevant. De rapportage gaat per definitie over iets in het universum van discussie en kan per definitie slechts gerapporteerd worden door iemand of iets in de omgeving zodra of nadat de situatie of gebeurtenis is waargenomen. Het is zeer goed mogelijk, en inderdaad in de praktijk vaak het geval, dat al enige tijd verstreken kan zijn voordat een gebeurtenis wordt waargenomen. In dat geval neemt men alleen de uit de gebeurtenis voortgekomen situatie waar. Ziet men het gebeuren zelf, dan kan vanzelfsprekend de situatie zoals die was voor de gebeurtenis, als de situatie na het plaatsvinden van de gebeurtenis bekend zijn. In INFOMOD wordt aangenomen, dat uit het gebeuren – het proces van veranderen – logischerwijs het resultaat – de resulterende situatie – volgt. Een en ander is natuurlijk ook afhankelijk van de situatie voor de gebeurtenis, maar dit ligt vaak in het eindresultaat verankerd: anders zou het eind-



Figuur 4.2: Globale architectuur van het informatiesysteem

schrijving, in het tweede geval van een situatie- (toestand) en transitie-georiënteerde beschrijving. In beide gevallen is de beschrijving (de rapportage) altijd na de waarneming en dus na de gebeurtenis, hoe kort deze tijd ook mag zijn. Daarbij komt dan nog, dat in vele gevallen alleen het resultaat het belangrijkste geacht wordt. In INFOMOD is dan ook voor een situatie- en transitie-georiënteerde beschrijving gekozen.

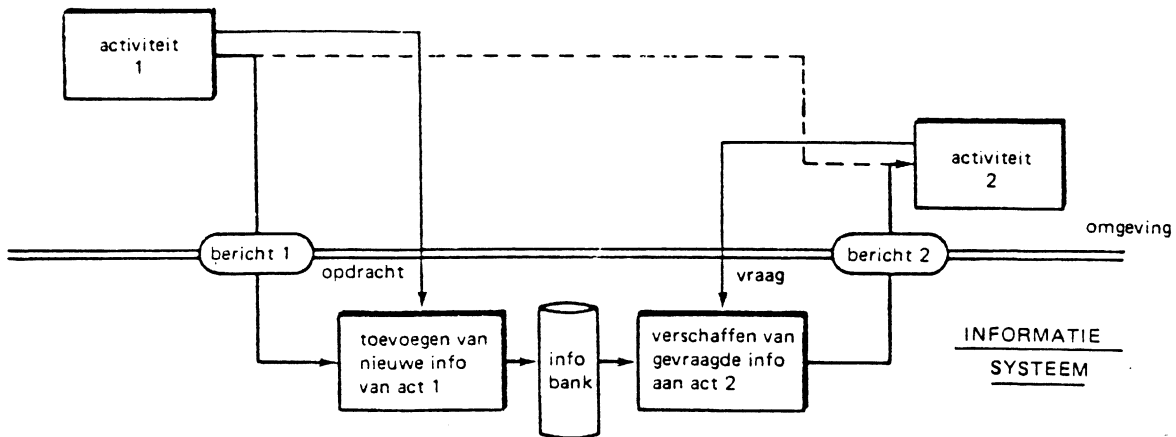
De rapportage over een situatie of gebeurtenis vindt plaats in de vorm van een bericht, dat aan het informatiesysteem wordt aangeboden. Dit heeft tot gevolg, dat in het informatiesysteem een actie moet plaatsvinden, die of de nieuwe informatie toevoegt aan de in het informatiesysteem aanwezige informatie, of het bericht op grond van regels verwerpt. Dit laatste vindt plaats als acceptatie van het bericht tot tegenspraak in de informatie zou leiden. Het streven is er immers op gericht, dat de informatie in het systeem zonder tegenspraak moet zijn. Eigenlijk zou men wensen, dat de informatie accuraat is,



Figuur 4.3: Bedrijfsactiviteiten, welke informatie uitwisselen

resultaat immers niet zo zijn. In het algemeen kan gesteld worden, dat het resultaat uit een beschrijving van het gebeuren afgeleid kan worden, maar even goed dat het gebeuren geconcludeerd kan worden uit een beschrijving van beginsituatie en eindresultaat. In het eerste geval spreekt men van een actie-georiënteerde be-

d. w.z. dat het ware situaties of gebeurtenissen beschrijft. Dit kan alleen vastgesteld worden door een gebruiker, die zowel de informatie als het universum van discussie zelf ziet of kent, en deze met elkaar kan vergelijken. Een informatiesysteem kan nooit de juistheid van informatie garanderen. Informatie, die in tegenspraak is, of niet aan



Figuur 4.4: Informatie-uitwisseling via het informatiesysteem

gestelde regels beantwoordt, is echter zonder meer onaccuraat, en voldoet dus zeker niet aan de betrouwbaarheidseis.

Ook de rapportage van informatie vanuit het systeem aan een gebruiker in de omgeving is het resultaat van een actie van het informatiesysteem. In dit geval wordt de actie vaak veroorzaakt doordat een vraag aan het systeem gesteld wordt.

INFOMOD kent nog een derde soort acties. Deze acties leiden informatie af uit reeds aanwezige informatie. Dit vindt plaats volgens van te voren gespecificeerde regels over het verband tussen de zaken, die door de informatie beschreven worden. Dergelijke acties worden onder andere gebruikt voor het 'automatisch nemen' van beslissingen, en voor allerlei controle en besturingstoepassingen. Voor alle drie de soorten acties geldt, dat zij alleen toelaatbaar zijn indien de resulterende informatie, zowel die in het systeem, als die welke gerapporteerd wordt, aan de integriteitseisen voldoet, d.w.z. zonder tegenspraak is.

De acties worden uitgevoerd door het verwerkingsmechanisme, waarbij de informatiebank de verzameling in het systeem aanwezige informatie vormt. Zo kan een globale architectuur weergegeven worden als in figuur 4.2.

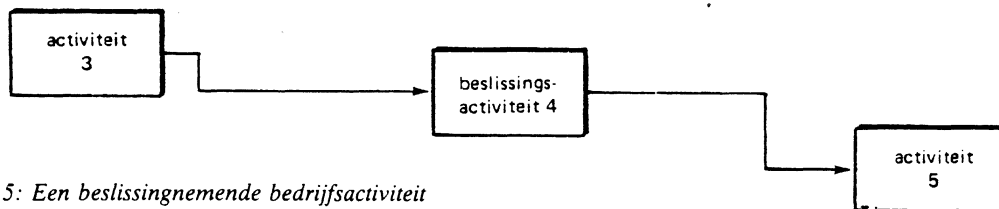
4.4 Informatie-uitwisseling met de Omgeving

Zoals in het begin gesteld, is het eerste doel van een informatiesysteem de uitwisseling van informatie tussen gebruikers. Gaan we uit van een bedrijfssituatie, dan is deze informatie nodig om de gebruikers in staat te stellen

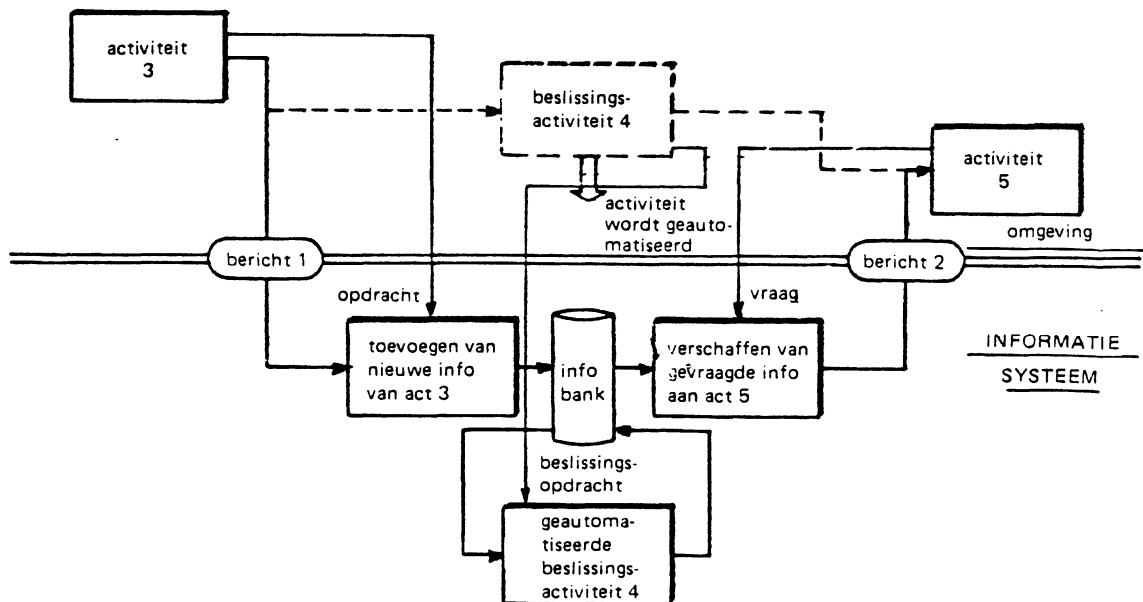
hun taken in de bedrijfsactiviteiten te verrichten. Deze informatie moet (uiteindelijk) verstrekt worden door die activiteiten, die verantwoordelijk zijn voor de situaties en veranderingen hierin, waarover de informatie handelt, of tenminste verantwoordelijk zijn voor de rapportage hierover. Met andere woorden, er bestaat een informatie-uitwisseling tussen deze bedrijfsactiviteiten. Dit is weergegeven in een model, gebruikmakend van SADT [11] technieken, als afgebeeld in figuur 4.3.

Een dergelijke informatie-uitwisseling kan direct plaatsvinden maar kan ook via een informatiesysteem uitgevoerd worden. In dit laatste geval zal de informatie verschaffende activiteit opdracht geven aan het informatiesysteem om een bericht met de betrokken informatie als inhoud te accepteren. De activiteit, die de informatie nodig heeft, kan deze opvragen uit het systeem (zie figuur 4.4).

Een andere mogelijkheid is, dat de toevoeging van de informatie aan het systeem automatisch de rapportage van de informatie aan de tweede activiteit tot gevolg heeft. Dit principe maakt het mogelijk, dat niet alle informatie, die de tweede activiteit nodig heeft, noodzakelijkerwijs door de eerste activiteit verstrekt dient te worden. In dezelfde zin is het ook niet noodzakelijk, dat alle informatie, die de eerste activiteit rapporteert, ook doorgegeven zou moeten worden aan de tweede activiteit. Dit zal in de praktijk dikwijls het geval zijn. Dit leidt natuurlijk wel tot een wat ingewikkelder communicatie-patroon tussen bedrijfsactiviteiten en acties van het informatiesysteem.



Figuur 4.5: Een beslissingnemende bedrijfsactiviteit



Figuur 4.6: Geautomatiseerde beslissingsactiviteit

4.5 Geautomatiseerde Beslissingsprocessen

Vaak is de enige taak van een bedrijfsactiviteit om informatie af te leiden uit reeds bekende informatie, of op grond hiervan beslissingen te nemen, die dan weer aan anderen medegedeeld wordt. In een aantal gevallen kunnen de overwegingen en regels hiervoor volledig geformaliseerd worden, d.w.z. de resulterende informatie kan geheel in termen van initiële informatie en regels uitgedrukt worden. Het is niet belangrijk of het verband gedefinieerd wordt, of dat beschreven wordt hoe het resultaat verkregen moet worden. Bedrijfsactiviteiten, waarvoor dit geldt, kunnen geautomatiseerd worden, hetgeen inhoudt, dat zij door het geautomatiseerde informatiesysteem uitgevoerd kunnen worden. Het principe hiervan is in de figuren 4.5 en 4.6 weergegeven.

In figuur 4.5 staat een situatie weergegeven, waarbij de activiteit 5, bijv. een verzendingsafdeling, die een levering verzendklaar maakt, informatie nodig heeft, bijv. welke order naar welke klant verzonden dient te worden. De beslissing hiertoe wordt genomen door activiteit 4 na controle of er voldoende voorraad van het bestelde product is. Deze activiteit zal echter pas plaatsvinden, nadat activiteit 3 – de verkoop – een bestelling heeft ontvangen van de klant.

Nemen wij aan, dat de voorraad controle en daaruitvolgende beslissing tot levering volledig formaliseerbaar is, dan kan de activiteit 4 door het informatiesysteem uitgevoerd worden – figuur 4.6. Deze geautomatiseerde activiteit gaat ervan uit, dat de benodigde informatie in het informatiesysteem aanwezig is. Er is dus een informatie-inbrengende actie 3 in het informatiesysteem nodig om de door de verkoopactiviteit 3 ontvangen bestelling in te brengen. Uiteraard dient de beslissing aan de verzendactiviteit 5 medegedeeld te worden. Hiertoe dient actie 5. In figuur 4.6 is er van uitgegaan, dat de gebruiker, die verantwoordelijk is voor de beslissing dat er geleverd mag worden, zelf de opdracht geeft aan het informatiesysteem de controle uit te voeren. Ook is aangenomen, dat de verzendactiviteit zelf nagaat, of er orders te verzenden zijn. Andere situaties zijn even goed denkbaar. Zo kan het inbrengen van informatie over een bestelling door activiteit 3 automatisch tot gevolg hebben, dat de controle en beslissing plaatsvindt, en dit vanzelf aan de verzendafdeling wordt doorgegeven.

Bovenstaande voorbeelden zijn de twee basisvormen van gebruik van een informatiesysteem. In de praktijk zullen vaak veel ingewikkelder combinaties van deze vormen voorkomen. Hierbij zullen de bedrijfsactiviteiten vaak in 'dialoogvorm' gebruik maken van de mogelijkheden van een informatiesysteem. Dergelijke bedrijfsactiviteiten zijn dan evenwel verder op te splitsen in deelactiviteiten, die noodzakelijkerwijs de intuïtie, keuzevrijheid en uitvoeringsmogelijkheden van de gebruiker vereisen, en die welke formaliseerbaar en dus automatiseerbaar zijn. Hierbij is een dergelijke activiteit dan uitgewerkt in een combinatie van activiteiten en informatiesysteemacties volgens het principe van figuur 4.6. Zuivere informatieuitwisseling tussen twee bedrijfsactiviteiten, waarbij dus geen nieuwe informatie ontstaat, zijn altijd uit te voeren met constructies gebaseerd op het principe van figuur 4.4.

5 DE BESCHRIJVING VAN EEN INFORMATIEMODEL

Een informatiemodel dient tenminste twee doeleinden:

- gebruikers te helpen begrijpen over welke dingen, situaties en gebeurtenissen de informatie in het informatiesysteem gaat;
- een nauwkeurige en volledige specificatie te geven welke informatie-uitwisselingen en verwerking mogelijk zijn en wat hierbij geldt.

Deze twee doelstellingen zijn vaak min of meer tegenstrijdig. Nauwkeurige en volledige specificaties, speciaal in een omvang zoals in het gemiddelde informatiemodel wordt aangetroffen, zijn vaak moeilijk te overzien en dus te begrijpen, zelfs voor een ervaren informatie-analyst. Trouwens, de eis, dat een informatiemodel verwerkbaar moet zijn op een computer, komt de leesbaarheid voor een gebruiker lang niet altijd ten goede. Grafische hulpmiddelen, zoals tekeningen, schema's e.d. zijn vaak uitermate geschikt om de lezer een overzicht te geven van een inzicht in ingewikkelde structuren. Deze kunnen dan toegelicht worden in informele beschrijvingen. Deze zijn echter lang niet altijd geschikt om op een computer verwerkt te worden, te meer omdat natuurlijke taal vaak dubbelzinnige beschrijvingen toelaat. Daarbij komt nog, dat diagrammen en schema's niet tegelijk nauwkeurig, volledig, en overzichtelijk kunnen zijn: te veel details camoufleren de essentiële zaken. Een overzicht wordt verkregen door juist de details weg te laten.

INFOMOD maakt daarom gebruik van twee manieren van beschrijven:

- grafische technieken voor het verstrekken van overzichten van ingewikkelde structuren, zoals de samenhang van zaken in situaties en gebeurtenissen, en de coördinatie van acties;
- een formele taal voor een nauwkeurige en volledige specificatie van de drie delen van een informatiemodel.

De formele taal van INFOMOD ligt dicht tegen de natuurlijke taal aan. Daarom ook kent INFOMOD verschillende versies. Naast de (oorspronkelijke) Engelse versie zijn op dit moment een Nederlandse, een Franse en een Portugese versie beschikbaar. Een Duitse en een Italiaanse versie zijn in voorbereiding. Alle versies zijn direct in elkaar vertaalbaar, aangezien zij op dezelfde syntax gebaseerd zijn.

De informatie-analyse en de beschrijving van een informatiemodel gaat in een aantal stappen. Deze zijn ruwweg in vier groepen te verdelen:

1. Ruwe schets:
Bepaal eerst in grote lijnen wat de belangrijkste gebruikersgroepen – bedrijfsonderdelen en bedrijfsactiviteiten – zijn waarvoor informatie nodig is, en wat hiervoor de belangrijkste zaken en omstandigheden zijn waarover zij informatie nodig hebben.
2. Informatiestructuurmodel:
Beschrijf vervolgens nauwkeurig alle dingen en zaken, die hierbij betrokken zijn, en hun exacte samenhang. Dit betreft dus niet alleen de hoofdzaken, die al bij de ruwe schets gevonden waren, maar ook alle bijzaken, die op een of andere manier relevant zijn.
3. Informatieregels:
Leg hierna alle regels vast, die omschrijven wat voor bovenstaande zaken en omstandigheden kan

of moet gelden. Hierbij komen vooral ook de samenhang in de tijd van de gebeurtenissen en mogelijke veranderingen aan bod.

4. Informatie-operatiespecificaties:
Bepaal de communicatiepatronen tussen de gebruikers en de daaruit volgende informatie-uitwisselingen met het informatiesysteem. Beschrijf voor deze laatste de inhoud van de bericht-uitwisseling, en leg de specificatie en condities vast van de daarbij nodige acties van het informatiesysteem. Ga na welke volgorde in de tijd een rol speelt.

In het kader van INFOMOD-toepassingen wordt voor de bepaling van de gebruikersgroepen – de bedrijfsactiviteiten – en de hiertussen nodige communicatiepatronen meestal gebruik gemaakt van de SADT, de Structured Analysis and Design Technique van SofTech [9].

6 DE RUWE SCHETS

Bij het opzetten van een informatiemodel (of informatie-submodel) zal een informatie-analyst eerst in grote lijnen bepalen wat de belangrijkste gebruikersgroepen – bedrijfsonderdelen en bedrijfsactiviteiten – zijn waarvoor informatie nodig is, en wat hiervoor de belangrijkste zaken en omstandigheden zijn waarover zij informatie nodig hebben. Hij maakt als het ware een ruwe schets van het universum van discussie en de omgeving van het informatiesysteem. Dit zou men kunnen vergelijken met een kunstschilder, die eerst een schets maakt waarin hij naderhand steeds meer details invult totdat het uiteindelijke schilderij voltooid is.

In deze fase van de informatie-analyse is de eerste doelstelling een beeld te krijgen van de belangrijkste dingen en zaken, waarover informatie nodig is. Zoals reeds vermeld, zijn dit de entiteiten in het universum van discussie. Hiervan wil of moet men allerlei eigenschappen weten, en in welke samenhang met andere entiteiten zij betrokken zijn. Bijvoorbeeld, van een medewerker Jan wil men weten dat hij voor de afdeling Verkoop werkt. In INFOMOD kan dit in een entiteit-associatiediagram worden weergegeven (figuur 6.1).



Figuur 6.1: Entiteit-associatiediagram

·Medewerker Jan werkt voor afdeling Verkoop.

In een dergelijk diagram worden de entiteiten aangeduid met cirkels, het verband tussen entiteiten – de associaties – met lijnen. In natuurlijke taal kan dit omschreven worden met:

Informatie-uitwisseling in natuurlijke taal is veelal onderwerp-georiënteerd. Dat wil zeggen, dat in overdracht van informatie het onderwerp centraal gesteld wordt, en dat van dit onderwerp eigenschappen en situaties, waarin het verkeert, beschreven worden. Deze onderwerpen zijn de dingen en zaken waarover men informatie uitwisselt. In dezelfde zin is INFOMOD entiteit-georiënteerd. Stellen wij in het bovenstaande voorbeeld Jan als centraal onderwerp, dan is 'werkt voor afdeling Verkoop' een situatie, die voor hem geldt, die aan hem toegevoegd wordt – een attribuut. Een attribuut is een eigenschap

van een entiteit, waarin mogelijk ook een of meer andere entiteiten betrokken kunnen zijn. Entiteiten en attributen zijn de basis-elementen in INFOMOD. Deze entiteiten zijn de individuele dingen en zaken, de attributen zijn hun individuele eigenschappen en omstandigheden.

Een informatiemodel is a.h.w. een blauwdruk van het universum van discussie. Hierin worden de entiteiten en attributen in meer gegeneraliseerde termen beschreven. Dit berust op classificatie, d.w.z. entiteiten worden op grond van eigenschappen en omstandigheden, die zij min of meer gemeen hebben ingedeeld in soorten: entiteit-typen. Dit is een zeer natuurlijke zaak, die dan ook in onze taal en taalgebruik ingebakken zit. Voor bovenstaand voorbeeld is een algemenere beschrijving, het patroon (zie figuur 6.2).



Figuur 6.2: 'Medewerker werkt voor afdeling'

De formele notatie van deze situatie in INFOMOD is:

MEDEWERKER (werkt voor afdeling)

Het gedeelte tussen haakjes duidt het attribuut aan van de entiteit, die zelf voor de haakjes vermeld wordt. In INFOMOD is het verschil tussen de situatie van figuur 6.1 en 6.2 alleen, dat in het eerste geval twee bepaalde entiteiten afgebeeld worden, terwijl in het tweede geval een patroon gegeven wordt, dat kan gelden voor een medewerker en een afdeling, welke dit ook mogen zijn.

Het is belangrijk zich er van bewust te zijn, dat hierbij de analyse op de dingen en hun omstandigheden gericht is, dus op het universum van discussie zelf. Hoe de informatie hierover beschreven wordt, hoe zij gebruikt wordt, en zelfs voor wie de informatie nodig is, is in deze fase van de analyse minder belangrijk. Toch zijn deze laatste zaken niet te verwaarlozen. De analyse van de gebruikersactiviteiten, waarvoor de informatie nodig is, hoe deze de informatie gebruiken, en hoe de informatie door hen verkregen wordt als (nog) geen gebruik gemaakt wordt van een geautomatiseerd informatiesysteem, kan waardevolle zaken over de entiteiten zelf en hun gedrag aan het licht brengen.

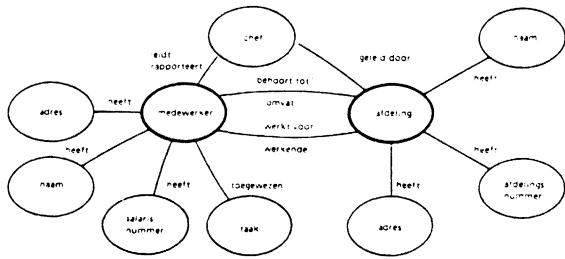
Als demonstratie van het resultaat van een ruwe schets wordt een model beschreven van de volgende situatie ten behoeve van het personeelsbeheer:

Personeelsbeheer (visie):

Een organisatie wil informatie hebben over het personeel in dienst bij de verschillende afdelingen. Afdelingen zijn bekend onder hun naam en afdelingsnummer. Zij worden geleid door chefs. Medewerkers zijn bekend onder hun naam en salarisnummer. Medewerkers behoren tot een afdeling. Medewerkers en afdelingen hebben adressen. Medewerkers hebben diverse taken toegewezen gekregen, indien zij voor een afdeling werken; dit kan hun eigen afdeling zijn of andere afdelingen. Zij rapporteren echter altijd aan de chef van hun eigen afdeling.

Deze situatiebeschrijving is reeds in generalizerende termen beschreven. Dit gebeurt in de praktijk zeer vaak: de

gebruiker geeft in de beschrijving van de bedrijfssituatie al aan hoe hij de zaken naar soorten indeelt. Een entiteit-associatiediagram van deze situatie is afgebeeld in figuur 6.3.



Figuur 6.3: Entiteit-associatie-diagram van de personeelsbeheersituatie

In deze figuur zijn de twee entiteiten medewerker en afdeling met zwaardere lijnen aangegeven. Deze entiteiten zijn degenen waarnaar de belangstelling in de eerste plaats uitgaat, waarover informatie nodig is. In INFOMOD worden dit primaire entiteiten genoemd. De andere entiteiten zijn slechts belangrijk voor zover zij een rol spelen in een situatie of eigenschap, de attributen, van de primaire entiteiten. De attributen van de primaire entiteiten, voor zover zij relevant zijn voor de visie van het personeelsbeheer worden in lijsten weergegeven. Deze lijsten hebben in INFOMOD een zgn. vectornotatievorm: de entiteitvector. Een entiteitvector is de lijst van de attributen zelf, die voor een individuele entiteit gelden. Zoals reeds opgemerkt, beschrijft een informatiemodel het patroon hiervoor in algemener gestelde termen. Hierin is de basisnotatievorm het entiteitvectorpatroon. Voor het voorbeeld zijn de volgende entiteitvectors belangrijk:

```
MEDEWERKER (naam, adres, salarisnummer,
             behoort tot afdeling, werkt voor afdeling,
             rapporteert aan chef, toegewezen taak)

AFDELING (naam, adres, afdelingsnummer,
          omvat medewerker, werkende medewerker,
          geleid door chef)
```

Zoals uit bovenstaand voorbeeld blijkt, kan een en dezelfde situatie vanuit ieder der betrokken entiteiten beschouwd en beschreven worden. Dit leidt soms in verschillen in omschrijving van de associatie in de attributen. Hierop wordt in de volgende paragraaf teruggekomen.

7 HET INFORMATIE-STRUCTUUR-MODEL

De doelstelling van het informatie-structuur-model is de nauwkeurige beschrijving van alle primaire entiteiten in hun onderlinge samenhang – de entiteiten-structuur, en al hun attributen die relevant geacht worden binnen de scope van het model. Voor iedere entiteitsoort – het entiteit-type – wordt een entiteitvector-patroon gespecificeerd, dat aangeeft welke attributen voor de entiteiten relevant zijn. Hierbij wordt uitgegaan van de ruwe schets. Aanvullende gesprekken met de gebruikers en verder gravende analyse zal de analyst in staat stellen meer en meer details in te vullen. Op deze wijze ontstaat langzamerhand het volledige beeld van de structuur van het universum van discussie. Hierbij zullen zeker nieuwe entiteitsoorten 'ontdekt' worden evenals nieuwe attributoorten. Het proces is een verfining in stappen, dat

(meestal) gelijk op gaat met de 'gestructureerde opsplitsing' van de bedrijfsactiviteiten bij de SADT functionele analyse.

De structuur tussen de entiteiten wordt bepaald doordat primaire entiteiten een rol als geassocieerde entiteit in het attribuut van een andere primaire entiteit spelen. In het voorbeeld van het personeelsbeheer is dit o.m. het geval bij MEDEWERKER en AFDELING. Een AFDELING is betrokken in de attributen MEDEWERKER (behoort tot AFDELING) en MEDEWERKER (werkt voor AFDELING). Dezelfde situaties worden echter ook vanuit de AFDELING bekeken. In dit geval kan de aanduiding 'behoort tot' van de associatie misverstanden opwekken: AFDELING (behoort tot MEDEWERKER) wordt waarschijnlijk anders begrepen. Dit ligt aan de taalinterpretatie. Een gebruiker-vriendelijke oplossing ligt echter voor de hand: Omschrijf de associatie op een tweede manier gezien vanuit het standpunt van de afdeling. Dit levert de attributen AFDELING (omvat MEDEWERKER) en AFDELING (werkende MEDEWERKER) op. Deze oplossing, twee omschrijvingen voor de zelfde situatie, moet in een formeel model echter wel aangegeven worden. Hiertoe kent INFOMOD de 'regel van correspondentie':

```
MEDEWERKER (behoort tot AFDELING)
correspondeert met
AFDELING (omvat MEDEWERKER)

MEDEWERKER (werkt voor AFDELING)
correspondeert met
AFDELING (werkende MEDEWERKER)
```

(In INFOMOD worden primaire entiteiten meestal met hoofdletters aangeduid).

Andere voorbeelden van met elkaar corresponderende attributen zijn die, waar de associatie vanuit de verschillende kanten gezien door een actieve en een lijdende vorm van een werkwoord worden aangegeven. 'Een chef leidt een afdeling', bijvoorbeeld, duidt dezelfde situatie, aan als 'een afdeling wordt geleid door een chef'. Indien de gekozen verwijzing naar de associatie geen misverstand kan veroorzaken, gezien vanuit welke deelnemende entiteit dan ook, dan kan zij in de corresponderende attributen dezelfde verwijzing naar de associatie gebezigd worden. In die gevallen is het ook niet nodig de regel van correspondentie expliciet te vermelden. Dit neemt echter niet weg, dat de regel impliciet wel geldt. Een voorbeeld hiervan is de toegewezen taak van een medewerker. Men zou hier zonder misverstand ook kunnen spreken van de toegewezen medewerker van een taak.

Een ander aspect, dat de structuur beïnvloedt, met name indien een model opgesteld wordt, is het feit, dat een entiteit vaak meer dan één attribuut van een bepaalde soort kan hebben. In INFOMOD worden deze multi-pele attributen genoemd. Dikwijls zijn deze juist relevant indien er andere primaire entiteiten bij betrokken zijn. In INFOMOD-diagrammen wordt de meervoudigheid aangegeven met aanduidingen 1 - n, en n - m (zie figuur 7.1),



Figuur 7.1: Meervoudige associaties tussen entiteiten.

terwijl in het entiteitvector-patroon de attribuutaanduiding tussen accoladen geplaatst wordt:

```
AFDELING ( (omvat MEDEWERKER) (werkende MEDEWERKER) )
MEDEWERKER ( behoort tot AFDELING ( werkt voor AFDELING) )
```

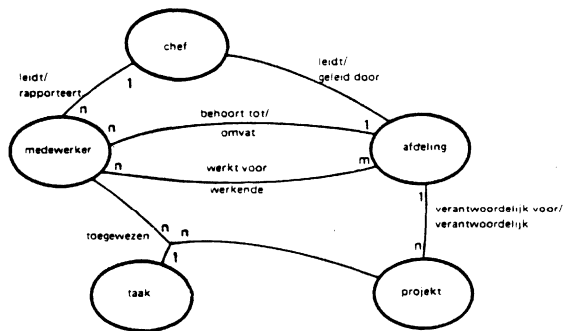
Om het resultaat van deze fase in het modelleerproces te demonstreren, nemen wij aan, dat de volgende additionele voor het personeelsbeheer relevante feiten aan het licht gekomen zijn.

Personeelsbeheer (aanvulling):

Van de chefs zijn de naam en salaris-nummer bekend. Ook hun adressen dienen geregistreerd te worden. Medewerkers en chefs verdienen een salaris. Van iedere afdeling moet het aantal medewerkers bekend zijn. De afdelingen zijn verantwoordelijk voor een aantal projecten. Medewerkers zijn belast met taken in een of meer van deze projecten. De geplande begindata, de geschatte duur en de verwachte einddata zijn eveneens belangrijk.

Omdat per primaire entiteitsoort alle belangrijke attributen gespecificeerd zijn in het entiteitvector-patroon, is het meestal gebruikelijk in deze fase alleen nog de associaties tussen de primaire entiteiten in een diagram weer te geven. Dit zijn de entiteit-structuurdiagrammen. Deze structuur tussen de belangrijkste entiteiten geeft met name een goed overzicht, en daarmee een illustratieve ingang tot het formele informatiemodel. Toevoeging van de secundaire entiteiten in de diagrammen komt dit overzicht niet ten goede, vooral niet bij uitgebreidere modellen.

Het resulterende entiteit-structuurdiagram voor het personeelsbeheer is weergegeven in figuur 7.2.



Figuur 7.2: Entiteit-structuurdiagram voor het Personeelsbeheer

Merk op dat de associatie tussen MEDEWERKER, PROJECT en TAAK een zgn. ternaire associatie is. In INFOMOD is geen modeltechnische beperking van het aantal entiteiten, dat in een associatie betrokken kan zijn. INFOMOD laat toe, dat de wereld gemodelleerd wordt, zoals zij waargenomen wordt. De regel van correspondentie, welke direct onder het vector-patroon van TAAK wordt aangetroffen, is hier uitgeschreven, om de regel bij ternaire situaties te demonstreren. Omdat de associatie-verwijzing hetzelfde is voor alle drie corresponderende attributen, hoeft de regel echter niet expliciet vermeld te worden (het mag wel).

Hieruit blijkt, dat twee in de ruwe schets reeds onderkende entiteitsoorten, TAAK en CHEF, nu ook als primair belangrijk geacht worden. Men wenst over hen informatie te hebben. Ook is een 'nieuwe' hoofdentiteitsoort aan het licht gekomen, het PROJECT. Het formele informatie-structuur-model wordt als volgt:

```

Informatie-model PERSONEELSBEHEER
Opmerking: Informatie-structuur-modelgedeelte

AFDELING (naam, adres, afdelingsnummer,
(omvat MEDEWERKER) (aantal-medewerkers)
(werkende MEDEWERKERS)
geleid door CHEF,
(verantwoordelijk voor PROJECT))

CHEF (naam, adres, salaris-nummer, verdient salaris,
leidt AFDELING, (leidt MEDEWERKER))

CHEF (leidt AFDELING)
correspondeert met
AFDELING (geleid door CHEF)

MEDEWERKER (naam, adres, salaris-nummer, verdient salaris,
behoort tot AFDELING, rapporteert aan CHEF,
(werkt voor AFDELING)
(toegewezen TAAK in PROJECT))

MEDEWERKER (behoort tot AFDELING)
correspondeert met
AFDELING (omvat MEDEWERKER)

MEDEWERKER (rapporteert aan CHEF)
correspondeert met
CHEF (leidt MEDEWERKER)

MEDEWERKER (werkt voor AFDELING)
correspondeert met
AFDELING (werkende MEDEWERKER)

PROJECT (verantwoordelijke AFDELING,
geplande begindatum, geplande duur, geplande einddatum,
(toegewezen TAAK aan MEDEWERKER))

PROJECT (verantwoordelijke AFDELING)
correspondeert met
AFDELING (verantwoordelijk voor PROJECT)

TAAK (geplande begindatum, geplande duur, geplande einddatum,
(toegewezen aan MEDEWERKER in PROJECT))

TAAK (toegewezen aan MEDEWERKER in PROJECT)
correspondeert met
MEDEWERKER (toegewezen TAAK in PROJECT)
correspondeert met
PROJECT (toegewezen TAAK aan MEDEWERKER)

Einde informatiemodel PERSONEELSBEHEER
  
```

8 INFORMATIEREGELS

Als de primaire entiteiten en hun attributen in kaart gebracht zijn, is de volgende stap de vaststelling welke regels over de samenhang en het mogelijke gedrag van de entiteiten gelden. Regels leggen per definitie vast wat geldt in het universum van discussie. Daarom kunnen zij gebruikt worden om de integriteit van de informatie te beheersen. Zijn geen of nauwelijks enkele regels gespecificeerd, dan is het niet goed mogelijk tegenspraken in de informatie 'geautomatiseerd' vast te stellen. Hoe meer regels bekend zijn, des te beter kan de integriteit bewaakt worden. Hiermee verkleint ook de kans dat de informatie desondanks niet juist is.

Regels worden in INFOMOD uitgedrukt in termen van entiteiten, hun attributen en betrekkingen daartussen. Sommige van deze regels leggen natuurwetten of algemeen aanvaarde normen vast, sommige zijn door de onderneming vastgesteld, andere zijn algemeen geldend in de maatschappij of gesteld door de overheid. De regels worden veelal gevonden door verdergaande analyse van de entiteiten en attributen in het universum van discussie. Hierbij komen vragen aan de orde als:

- Hoe identificeren we een entiteit?
- Moeten wij een attribuut onder alle omstandigheden kennen?
- Waarom heeft een entiteit dit attribuut?
- Is het attribuut afhankelijk van het voorkomen

van andere attributen van deze of andere entiteiten?

- Welke mogelijke attributen van het type kunnen voorkomen en onder welke omstandigheden?
- Tot welke soorten kan een entiteit behoren en onder welke omstandigheden?
- Hoe kan het attribuut in de loop van de tijd veranderen, onder welke omstandigheden gebeurt dit en beïnvloedt dit ook andere attributen van deze of andere entiteiten?

Zoals reeds opgemerkt worden de regels in de eerste plaats vastgesteld door bestudering van het universum van discussie zelf. Ze hebben immers hierop betrekking. Toch kan de bestudering en analyse van hoe informatie hierover door gebruikers en programmatuur in bestaande dataverwerking geproduceerd en verwerkt wordt, waardevolle aanwijzingen geven. Hierbij dient men zich echter wel te bedenken, dat dergelijke regels over de betekenis van de informatie in applicatieprogrammatuur begraven kunnen zijn in vele procedurele stappen, of soms in het geheel niet geïmplementeerd werden.

Diverse soorten regels kunnen onderkend worden. Het zou te ver voeren om alle mogelijkheden hier te bespreken. Daarom zal volstaan worden met een aantal representatieve voorbeelden aan de hand van het voorbeeld voor het personeelsbeheer.

Een van de meest voor de hand liggende regels is de identificatie van entiteiten. In INFOMOD wordt onder identificatie verstaan, dat een entiteit door een of meer attributen zich per definitie uniek onderscheidt van andere entiteiten. In de praktijk leggen deze attributen meestal een verband tussen de entiteiten en daartoe geschikte namen, nummers of andere codes. Een voorbeeld, in INFOMOD-notatie, is:

Identifice MEDEWERKER (naam, adres)
MEDEWERKER (salarisnummer)

In dit voorbeeld worden voor een entiteit MEDEWERKER twee mogelijke identificaties gespecificeerd, de combinatie van naam en adres, en apart hiervan het salarisnummer. Meestal wordt een dergelijke identificatiespecificatie in het informatiemodel direct onder het entiteitvector-patroon geschreven.

De aanduiding van attributen in een entiteitvector-patroon geeft in INFOMOD slechts aan, dat een entiteit van het betrokken type dergelijke attributen kan hebben, maar niet noodzakelijkerwijs zal hebben. Daarbij komt nog, dat ook al heeft de entiteit in de werkelijkheid het attribuut, dit nog niet automatisch inhoudt, dat dit in het informatiesysteem bekend hoeft te zijn. Wil men die garantie wel, dan dient een regel geformuleerd te worden. Naar aanleiding van deze regel kan het informatiesysteem dan controleren, of het attribuut bekend is zodra en zolang de entiteit bekend is aan het systeem. Dit is de regel van de vereiste attributen. Een voorbeeld is:

Vereist AFDELING (afdelingsnummer)

Een ander voorbeeld is:

Vereist MEDEWERKER (naam en adres of salarisnummer)

In dit voorbeeld wordt vereist dat van een medewerker één van beide identificaties bekend moet zijn. Vereisen wij beide gelijktijdig, dan kan informatie over de medewerker pas geregistreerd worden, nadat bijv. de salaris-

afdeling het salaris-nummer bepaald heeft. Dit kan bij sommige instanties wel enige tijd vergen, en gedurende die tijd zou geen enkele informatie over de medewerker via het informatiesysteem uitgewisseld kunnen worden, hoewel we precies weten wie bedoeld wordt.

In vele gevallen hangt het bestaan van een attribuut samen met dat van andere attributen. Een van de mogelijkheden is bijvoorbeeld, dat een medewerker zonder twijfel een salaris verdient, indien hij tot een afdeling behoort:

MEDEWERKER (behoort tot AFDELING)
impliceert voor een salaris
MEDEWERKER (verdient salaris)

Uit deze regel mag uiteraard niet het omgekeerde geconcludeerd worden: als de medewerker een salaris verdient, volgt hier nog niet uit, dat hij dus tot een afdeling behoort. De samenhang kan echter ook complexer zijn, zoals bijvoorbeeld voor de eis, dat medewerkers werken voor die afdelingen, waarvoor zij taken verrichten:

MEDEWERKER (toegewezen TAAK in PROJECT)
en
PROJECT (verantwoordelijke AFDELING)
impliceert
MEDEWERKER (werkt voor AFDELING)

Een ander voorbeeld legt vast dat het attribuut onlosmakelijk afhankelijk van andere attributen. Dit is het geval van de medewerker, die rapporteert (rapporteren moet) aan de chef van de afdeling waartoe hij behoort:

MEDEWERKER (rapporteert aan CHEF)
hangt af van
MEDEWERKER (behoort tot AFDELING)
en
AFDELING (geleid door CHEF)

Soms hangen twee attributen zeer nauw samen, maar kan men toch niet zeggen, dat de ene afhankelijk van de andere. Stel de situatie, dat een afdeling waartoe een medewerker behoort ook de afdeling is die hem betaalt:

MEDEWERKER (behoort tot AFDELING)
indient en alleen maar indien
MEDEWERKER (betaald door AFDELING)

Hierbij is het al veel lastiger aan te wijzen wat de oorzaak is en wat het gevolg. Behoort hij tot de afdeling omdat die hem betaalt, of is het andersom?

Veelal bestaat er een 'rekenkundig' verband tussen attributen. Een dergelijke samenhang kan gebruikt worden om een attribuut te berekenen, indien de andere bekend is. In ons voorbeeld kan de samenhang tussen begin en einddata van een project of een taak aldus vastgelegd worden:

TAAK (geplande einddatum =
TAAK (geplande begindatum) + TAAK (geplande duur))

Merk op, dat dergelijke regels weliswaar ten grondslag liggen aan berekenings- en afleidingsprocessen, die automatiseerbaar zijn, maar dat zij ook veelal gebruikt kunnen worden om informatie tegen tegenspraak te behoeven. Immers, indien de data en de duur van een taak aan het informatiesysteem bekend gemaakt worden – het systeem wordt niet gebruikt om deze vast te stellen – dan kan door controle gemakkelijk een conflict tussen de data ontdekt worden. Conflicten kunnen veel gemakkelijker ontstaan in ingewikkelder situaties, waarbij verschei-

dene entiteiten betrokken zijn. Het is bijvoorbeeld niet mogelijk, dat een project beëindigd is voordat alle daarbij betrokken taken voltooid zijn. Mocht desondanks de planninginformatie het tegendeel beweren, dan is er sprake van een tegenspraak in de informatie. M.a.w.:

```
PROJECT (geplande einddatum =
voor iedere TAAK
waarvoor TAAK (toegewezen is PROJECT
max. TAAK (geplande einddatum))
```

In dit voorbeeld zijn de attributen van een groep entiteiten betrokken. In INFOMOD kan men een groep specificeren door de formulering van een groepsselectie 'Voor ... waarvoor ...'. Het daarop volgende 'max: ...' is de specificering van een functie over de groep - in dit geval duidt het de laatste geplande einddatum aan.

Het ontstaan of de vaststelling van een attribuut kan gebaseerd zijn op een reeds bestaande situatie. Dit kan dan aanleiding zijn voor een beslissingsregel voor dit attribuut. Dergelijke beslissingsregels worden in INFOMOD vaak uitgedrukt in de vorm van een functie-specificatie: d.w.z. de regel voor het (nieuwe) attribuut wordt geformuleerd als functie van de entiteiten en attributen, die hierbij betrokken zijn. Een voorbeeld is de definiëring van de bepaling van een salaris-nummer. Laten wij aannemen, dat hiervoor geldt, dat iedere nieuwe medewerker een salaris-nummer krijgt, dat 1 hoger is dan de tot nog toe uitgegeven nummers. De formulering van de regel in ons voorbeeld luidt:

```
Nieuw-salaris-nummer: MEDEWERKER =
1 + voor iedere MEDEWERKER#
waarvoor MEDEWERKER# = MEDEWERKER
max: MEDEWERKER# (salaris-nummer)
en voor MEDEWERKER#
waarvoor MEDEWERKER# (salaris-nummer onbekend)
neem aan: MEDEWERKER# (salaris-nummer = 0)
```

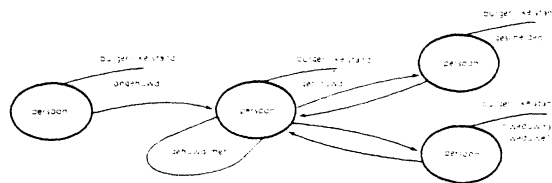
Hierbij is de functie-naam en het formele argument voor het '='-teken aangegeven. Achter het '='-teken is de functie-formulering zelf gespecificeerd.

In deze formulering wordt naar het begrip 'onbekend' gerefereerd. In INFOMOD is het begrip 'onbekend' gedefinieerd. Als in een uitdrukking een onbekende entiteit of attribuut betrokken is, dan is het resultaat van de uitdrukking in het algemeen ook onbekend. Als dus bijvoorbeeld in een beslissings situatie een entiteit of attribuut onbekend blijkt, zou de uitslag van de beslissing ook onbekend zijn. INFOMOD kent echter een functie 'neem aan'. Deze definieert welke situatie of welk attribuut aangenomen moet worden voor een uitdrukking of regel, indien zich een bepaalde gespecificeerde situatie voordoet. Hier is dat de situatie waarbij nog geen medewerkers met een salaris-nummer bekend zijn.

niert welke situatie of welk attribuut aangenomen moet worden voor een uitdrukking of regel, indien zich een bepaalde gespecificeerde situatie voordoet. Hier is dat de situatie waarbij nog geen medewerkers met een salaris-nummer bekend zijn.

Bovenstaande voorbeelden hadden alle betrekking op situaties, die voor entiteiten konden gelden, ongeacht hoe deze situatie tot stand gekomen was, of wat eraan voorafging. Verandering van situatie is echter even goed aan regels onderworpen. Hierbij ligt vaak de nadruk op welke situaties kunnen ontstaan indien een bepaalde situatie geldt. Bijvoorbeeld, een persoon kan slechts scheiden, indien hij of zij gehuwd is. Huwen kan slechts indien hij

of zij nog niet gehuwd is. In INFOMOD worden zulke verbanden tussen opeenvolgende situaties weergegeven in transitie-diagrammen - zie figuur 8.1.



Figuur 8.1: Transitie-diagram voor de burgerlijke staat van een persoon

Hierbij wordt meestal de situatie gekenschetst door een status. In het voorbeeld is dit uitgedrukt door de burgerlijke stand:

```
PERSON (burgerlike-stand = ongetrouwd
getrouwd
gescheiden
weduwe/weduwenaar)
```

De situatie zelf wordt natuurlijk door andere attributen van de betrokken entiteit bepaald, eventueel in samenhang met andere entiteiten. Bijvoorbeeld:

```
PERSON (burgerlike-stand = getrouwd)
impliceert
voor ieder PERSON#
PERSON (getrouwd met PERSON#)
```

(De aanduiding '#' geeft aan, dat het hier om een ander persoon gaat, hoewel deze tot dezelfde entiteit-soort PERSON behoort.)

Zo kan voor 'ongetrouwd' gesteld worden:

```
PERSON (burgerlike-stand = ongetrouwd)
impliceert
voor ieder PERSON#
niet PERSON (getrouwd met PERSON#)
```

De eis, dat iemand ongetrouwd moet zijn om te kunnen trouwen, luidt dan:

```
PERSON (burgerlike-stand = ongetrouwd)
gaat onmiddellijk voort aan
PERSON (burgerlike-stand = getrouwd)
```

In werkelijkheid is dit natuurlijk niet de enige mogelijke situatie die aan een huwelijk vooraf gaat. Gescheiden personen mogen immers ook hertrouwen:

```
PERSON (burgerlike-stand = ongetrouwd)
of anders
PERSON (burgerlike-stand = gescheiden)
of anders
PERSON (burgerlike-stand = weduwe/weduwenaar)
gaat onmiddellijk voort aan
PERSON (burgerlike-stand = getrouwd)
```

Andere regels handelen weliswaar niet over een mogelijke verandering, maar berusten wel op het feit, dat veranderingen optreden. Zij hebben betrekking op entiteiten en attributen in twee opeenvolgende situaties.

Een belangrijke groep regels zijn de 'wens' regels - regels die een gewenste situatie beschrijven. Dergelijke regels worden niet gebruikt om de integriteit van de informatie in het informatiesysteem te beheersen. Zij zijn noodzakelijk indien men van de mogelijkheden van het geautomatiseerde informatiesysteem gebruik wenst te maken bij de bewaking en besturing van het gedrag van zaken, gebeurtenissen en activiteiten in het universum van discussie of de omgeving. Overtreding van dit soort regels leidt niet tot niet-opnemen van de informatie in het sys-

teem, maar tot een reactie van het systeem naar buiten toe, bijvoorbeeld door een uitgaand waarschuwingsbericht. Dit bericht waarschuwt de gebruiker over de ongewenste, maar duidelijk mogelijke situatie. Een voorbeeld van een dergelijke regel is de stelling, dat een afdelingsbudget tenminste toereikend moet zijn om de salarissen van de medewerkers uit te betalen. Indien gemeld wordt, dat het budget ontoereikend is, zou dit beslist niet in het informatiesysteem geweigerd moeten worden. Wordt dit wel geweigerd, dan kan het systeem niet gebruikt worden om informatie over dit feit te verstrekken. De formulering van deze regel luidt:

Gewenst AFDELING (budeet) >=
 voor iedere MEDEWERKER
 waarvoor MEDEWERKER behoort tot AFDELING
 som MEDEWERKER (verdiensalaris)

In principe kan iedere soort regel zowel dienen als integriteitsregel of als wensregel. Het grammaticale verschil is het voorvoegsel 'gewenst'. Het volgende overzicht kan dienen om het onderscheid te verduidelijken:

soort regel	situatie in het UvD	informatie in het systeem
integriteit	(on)mogelijk	(on)toelaatbaar
wenselijkheid	(on)gewenst, maar mogelijk	toe te laten, wel/geen waarschuwing

Met name de analyse van de samenhang van zaken, omstandigheden en gebeurtenissen, en de modellering hiervan in geschikte regels maakt informatie-analyse tot een niet gemakkelijke aangelegenheid. Toch is dit noodzakelijk indien wij de inhoudelijke samenhang en betrouwbaarheid van informatie in onze informatiesystemen willen beheersen.

9 INFORMATIE-OPERATIES-SPECIFICATIES

De informatie-operaties-specificaties omvatten de specificaties van:

- de berichten, welke tussen de gebruikers in de omgeving en het informatiesysteem worden uitgewisseld,
- de voorwaarden en resultaten van de acties in het informatiesysteem, welke deze berichten verwerken,
- de volgorde in de tijd en de coördinatie van deze acties.

Als zodanig vormen zij, op het informatieniveau, de beschrijving van het interface tussen gebruikers en het informatiesysteem. Zij beschrijven hierbij niet hoe de acties de informatie opslaan, verwerken of naar buiten brengen, noch welke informatieregels daarbij gehandhaafd dienen te worden. De operaties-specificaties beperken zich tot welke informatie op welk moment aangeboden kan worden, c.q. naar aanleiding van welke criteria verstrekt dient te worden.

Het functionele model van de gebruikersactiviteiten en hun onderlinge communicatiepatroon is, naast het resultaat van de analyse van universum van discussie in het structuurmodel en de regels, uitgangspunt voor de informatie operaties-specificaties. De inhoud van de berichten wordt bepaald door de behoeften aan informatie-uit-

wisseling tussen de gebruikersactiviteiten en wordt uitgedrukt in termen van de attributen der relevante entiteiten. Dit wordt gespecificeerd in berichtvector-patronen. De scope en doelstelling van de acties wordt door deze berichten bepaald. De tijdsaspecten en volgorde van de bericht-uitwisselingen bepalen volgorde en coördinatie van de acties. Een voorbeeld van de rapportage van nieuwe informatie vanuit de omgeving aan het informatiesysteem is de mededeling, dat een nieuwe medewerker in dienst getreden is. We nemen aan dat de inhoud van de mededeling volgens het volgende berichtvector-patroon is:

NIJWE-MEDEWERKER (MEDEWERKER (naam, adres),
 AFDELING (naam))

Zoals dit voorbeeld laat zien, wordt de inhoud uitgedrukt in entiteit-subvector-patronen.

De aanbieding van een dergelijk bericht is de aanleiding tot een actie van het informatiesysteem, die de inhoud van het bericht als nieuwe informatie toevoegt aan het systeem. In het voorbeeld is aangenomen, dat het bericht dus onderdeel van de opdracht tot de actie zal zijn - de actuele parameter. De specificatie van deze (toelaatbare) actie luidt:

Start VOEG-NIJWE-MEDEWERKER-TOE- NIJWE-MEDEWERKER

Opmerking: Registreert naam en adres van een nieuwe medewerker als gegeven in het bericht NIJWE-MEDEWERKER en registreert tevens de toekenning van een salaris-nummer

Actie-omstand: Na ontvangst van de opdracht

Voeg toe MEDEWERKER (naam =
 NIJWE-MEDEWERKER (MEDEWERKER (naam),
 adres =
 NIJWE-MEDEWERKER (MEDEWERKER (adres)),
 salaris-nummer =
 Nieuw salaris-nummer MEDEWERKER
 behoort tot AFDELING =
 NIJWE-MEDEWERKER (AFDELING))

Indien dubbelaten
 dan uitgaande WISSELINGSBERICHT (MEDEWERKER (naam, adres),
 is al bekend.)

Concludeer

De 'Voeg toe' actiespecificatie geeft aan welke attributen van de entiteit aan het informatiesysteem bekend gemaakt worden. Drie van hen worden d.m.v. het bericht aangeboden. Het salaris-nummer wordt volgens de reeds gespecificeerde beslissingsregel vastgesteld. De nieuw toegevoegde informatie moet aan alle gestelde regels voldoen. Dit houdt in dit geval in, dat de combinatie naam-adres uniek moet zijn, het is immers de identificatie. Indien niet aan deze regel wordt voldaan, dan zal de actie geweigerd worden. Volgens bovenstaande specificatie wenst de gebruiker in dat geval hiervan op de hoogte gebracht te worden. Merk op, dat in deze specificatie niet gesproken wordt over het corresponderende attribuut AFDELING (omvat MEDEWERKER). Ook is geen voor databases gebruikelijke controle of de afdeling wel bekend is gespecificeerd. Binnen het kader van INFOMOD zijn dit twee impliciete zaken. D.w.z. corresponderende attributen zijn impliciet bekend, zodra een van hen bekend is. Ook geldt binnen INFOMOD, dat de afdeling, mocht zij nog niet bekend zijn, na de actie wel bekend is, ook al is dat dan alleen de naam en het feit dat de medewerker er toe behoort. Hoe dit binnen computerprogrammatuur verwezenlijkt wordt, is ontwerp en implementatie, en valt derhalve buiten de specificatie.

Als voorbeeld van een rapportage aan een gebruiker in de omgeving zal de melding van het salaris-nummer van

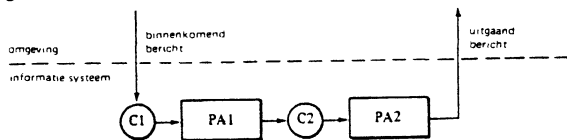
de nieuwe medewerker gedemonstreerd worden. De vraag kan natuurlijk alleen beantwoord worden, indien deze actie na de vorige actie plaatsvindt. Vergelijk ook het principediagram van figuur 4.4.

```

Start RAPPORTEER-SALARIS-NUMMER: MEDEWERKER (naam, adres)
Opmerking: Rapporteer het salaris-nummer van de medewerker
           waaraan naam en adres als argument gegeven is
Actie-conditie: Na ontvangst van de opdracht
Maak bekend MEDEWERKER (salaris-nummer)
           met MEDEWERKER (naam, adres)
Indien MEDEWERKER onbekend
• dan uitgaand BERICHT (MEDEWERKER (naam, adres)
           is onbekend.)
Consolideer

```

Zoals reeds in 4 opgemerkt, kunnen acties ook automatisch gekoppeld worden, d.w.z. het plaatsvinden – in dit voorbeeld – van de toevoeging van een nieuwe medewerker wordt automatisch gevolgd door de rapportage van het toegekende salaris-nummer. Dit is in het coördinatie-diagram van figuur 9.1 weergegeven (Vergelijk ook figuur 4.4).



Figuur 9.1: Aanmelding van nieuwe medewerker

De specificaties van beide acties dienen iets gewijzigd te worden. Dit is vetgedrukt aangegeven. (De mogelijkheid van een onbekende medewerker doet zich hier niet voor.)

```

Start VOEG-NIEUWE-MEDEWERKER-TOE: NIEUWE-MEDEWERKER
Opmerking: Registreert naam en adres van een nieuwe medewerker
           als gegeven in het bericht NIEUWE-MEDEWERKER en regi-
           streert tevens de toekenning van een salaris-nummer.
Actie-conditie: Na ontvangst van de opdracht
Voeg toe MEDEWERKER (naam =
           NIEUWE-MEDEWERKER (MEDEWERKER (naam)),
           adres =
           NIEUWE-MEDEWERKER (MEDEWERKER (adres)),
           salaris-nummer =
           Nieuw-salaris-nummer: MEDEWERKER,
           behoort tot AFDELING =
           NIEUWE-MEDEWERKER (AFDELING)).
Indien duplicaten
dan uitgaand WEIGERINGSBERICHT (MEDEWERKER (naam, adres),
           is al bekend.)
Vervolg met RAPPORTEER-SALARIS-NUMMER: MEDEWERKER.
Consolideer
Start RAPPORTEER-SALARIS-NUMMER: MEDEWERKER.
Opmerking: Rapporteer het salaris-nummer van de medewerker
           die als argument gegeven is.
Actie-conditie: Direct na VOEG-NIEUWE-MEDEWERKER-TOE.
           Maak bekend MEDEWERKER (salaris-nummer).
Consolideer

```

Hierbij kunnen nog de volgende kanttekeningen gemaakt worden. In de rapportage-actie hoeft het selectie-criterium niet expliciet vermeld te worden, als de betrokken entiteit al impliciet bekend is. Aangezien in beide rapportagevoorbeelden de betrokken entiteit al als parameter van de actie gespecificeerd is, is deze dus impliciet bekend. In het eerste geval is de entiteit bekend, omdat het bericht een volledige identificatie bevat. Het selectie-criterium 'met MEDEWERKER (naam, adres)' is toch vermeld als voorbeeld, hoewel dus niet echt nodig. Impliciete criteria mogen in INFOMOD echter altijd ook expliciet vermeld worden. In het tweede geval is het argument door de toevoegingsactie bepaald.

De uitwisseling is in twee toelaatbare acties beschreven, omdat het resultaat van de eerste actie een toelaatbare situatie in het informatiesysteem vastlegt, ongeacht of de rapportage volgt of niet. Bij implementatie kunnen beide acties natuurlijk zeer goed in een enkel programma uitgevoerd worden. Zeker bij ingewikkelder gevallen dient het dan toch beslist aanbeveling hiervoor in het enkele programma twee programmastappen (integriteits-eenheden) te onderkennen. In het algemeen kan trouwens gesteld worden, dat de toelaatbare actie in een INFOMOD operatiespecificatie aanleiding geeft tot implementatie door middel van een integriteitseenheid. Deze kan het enkele onderdeel van een programma zijn, maar ook een van de vele programmastappen (transactiestappen) in een programma [2]. Dit is een keuze tijdens het ontwerp van het informatiesysteem en in principe irrelevant voor de gebruiker. Hetgeen in het algemeen onder een transactie wordt verstaan, zal in INFOMOD in de regel een verzameling gecoördineerde toelaatbare acties zijn.

Het volgende voorbeeld laat de mogelijkheid zien van het rapporteren van een samengesteld bericht. Tevens demonstreert dit voorbeeld een schijnbaar spontane actie van het informatiesysteem, d.w.z. een actie, die periodiek plaatsvindt op vooraf bepaalde tijdstippen. Het personeelsbeheer heeft iedere maand op de twintigste een lijst nodig, waarop per afdeling vermeld wordt, welke salarissen uitbetaald dienen te worden, en welk bedrag hiermee gemoeid is. De specificatie van dit samengestelde bericht luidt:

```

SALARIS-LIJST
(maand, jaar,
 (AFDELINGS-SALARISSEN
 (AFDELING (naam, afdelingsnummer),
 (MEDEWERKER (naam, adres, salaris-nummer, verdient salaris),
 betaalbaar afdelings-totaal)),
 betaalbaar totaal)

```

De SALARIS-LIJST heeft drie attributen van zichzelf – maand, jaar, en betaalbaar bedrag. Het vierde – multiple – attribuut is een groep 'sub'-berichten, voor iedere betrokken afdeling één. Elk van deze bevat de naam en adres van de betrokken afdeling, een lijst van de medewerkers en het betaalbare bedrag per afdeling. Tenslotte worden van iedere medewerker de aangegeven attributen gerapporteerd. Merk op, dat normaliter geen bericht-opmaak in een informatie-model gespecificeerd wordt. De INFOMOD-taal maakt dit echter wel mogelijk. De betrokken actie-specificatie luidt:

```

Start MAANDELIJKE-SALARIS-LIJST
Opmerking: Rapporteer op de twintigste van iedere maand de uit-
           te betalen salarissen per afdeling, het totaal
           salaris-bedrag per afdeling, en het totaal generaal
           bedrag.
Actie-conditie: Voor iedere maand:
           op de 20ste dag in maand.
Uitgaand SALARIS-LIJST
SALARIS-LIJST (maand = deze maand, jaar = dit jaar)
Voor iedere AFDELING geldt:
(SALARIS-LIJST (AFDELINGS-SALARISSEN (AFDELING))
Voor iedere MEDEWERKER
waarvoor MEDEWERKER (behoort tot AFDELING) geldt:
SALARIS-LIJST
(AFDELINGS-SALARISSEN (MEDEWERKER))
SALARIS-LIJST
(AFDELINGS-SALARISSEN (betaalbaar afdelings-totaal =
voor iedere MEDEWERKER
waarvoor MEDEWERKER (behoort tot AFDELING)
som: MEDEWERKER (verdiend salaris))
SALARIS-LIJST (betaalbaar totaal =
som: SALARIS-LIJST
((AFDELINGS-SALARISSEN (betaalbaar afdelings-totaal))
Consolideer

```

In dit voorbeeld komt een aantal groepsspecificaties – 'Voor iedere .. geldt' – voor. Deze kan zich uitstrekken over een enkel actie-specificatieonderdeel – Voor iedere MEDEWERKER geldt: Een dergelijke groepsspecificatie kan echter ook voor een aantal actie-specificatie onderdelen gelden – Voor iedere AFDELING geldt: { }.

10 SLOTOPMERKINGEN

Het bovenstaande heeft alleen een indruk willen geven van hetgeen INFOMOD bieden kan. Een uitputtende behandeling van alle mogelijkheden en moeilijkheden zou echter te ver gaan voor dit artikel.

11 REFERENTIES

- 1 ANSI/X3/SPARC. 'Study Group on Data Base Management Systems: Interim Report 75-02-08'. In: ACM SIGMOD Newsletter. FDT. Vol. 7. No 2. 1975.
- 2 Corporate ISA-TMS. 'Multi-DBDC Rules'. Philips Corporate ISA. TMS/MW publication Ctn. 4322 270 09441. 1983.
- 3 CSD-ISA 'Principles for Definition of Data Elements'. Corporate ISA-TMS publication UT-D 1508/02. Dec. 1982.
- 4 Date, C. J. 'An Introduction to database systems'. 3e ed., Addison-Wesley. 1981.
- 5 van Griethuysen, J. J. (ed.) 'Concepts and Terminology for the Conceptual Schema en the Information Base', ISO/TC97/SC5 – N 695. ANSI. New York, 1982.
- Zie ook: Griethuysen, J. J. van: 'Grondslagen en terminologie voor het conceptuele schema'. In: *Informatie*, jrg. 23, nr. 7/8 (juli/augustus 1981) blz. 466-475.
- 6 Van Griethuysen, J. J., Jardine, D. A., 'The INFO-MOD Approach to Information Modelling'. Philips Corporate ISA-TMS publication Ctn 4322 270 09661-09671-09681 (UDR-TMS/SM-83/131:133). June 1983.
- 7 Van Griethuysen, J. J., van Melis, W. H. P. M., van Riel, A. J. J., 'Data organization in DB/DC-systems', Philips Corporate ISA-TMS publication Ctn. 4322 270 07881. 1981.
- 8 Martin, J. 'An Information Systems Manifesto', SA-VANT Research Studies, 1983.
- 9 Meersman, R., Ponsaert, F., de Troyer, O., 'The RIDL Language user and reference manual'. ICIAS Report, Control Data, 1984.
- 10 Nijssen, G. M., 'A framework for advanced mass storage applications', In: Medinfo 80, Proceedings of the Third World Conference on Medical Informatics, Tokio, North-Holland Publishing Company, 1980.
- Zie ook: Verheyen, G. M. A. en J. van Bekkum, 'Nijssen's Informatie-Analyse Methode (NIAM)'. In: *Informatie*, jrg 24, nr. 1 (maart 1982) blz. 3-16.
- 11 Ross, D. T., Dickover, M. E., McGowan, C. L., 'Structured Analysis and Design Technique SADT', Sof-Tech Report. 1976.
- Zie ook: Maarssen, ing. L. A. and C. L. McGowan Ph.D. 'Structured Analysis and Design Technique (SADT)' In: *Informatie*, jrg. 23, nr. 7/8 (juli/augustus 1981), blz. 433-442.

POLY-AUTOMATISERINGS ZAKBOEKJE

Begin mei is verschenen het Poly-automatiserings Zakboekje, een nieuwe uitgave van Koninklijke PBNA, instelling voor schriftelijk onderwijs te Arnhem.

Het Poly-automatiserings Zakboekje vormt een onderdeel van de reeks Poly-zakboekjes. In deze reeks zijn reeds verschenen het Polytechnisch Zakboekje, het Polysociaal Zakboekje en het Poly-economisch Zakboekje.

Het Poly-automatiserings Zakboekje omvat ruim 1200 pagina's met compacte informatie en gegevens met betrekking tot alle aspecten van de vakgebieden automatisering en informatica.

Het Poly-automatiserings Zakboekje is samengesteld door een redactiecommissie bestaande uit prof. dr. Th. M. A. Bemelmans, hoogleraar Bestuurlijke Informatiesystemen en Automatisering, Technische Hogeschool Eindhoven, prof. dr. J. A. van der Pool, buitengewoon hoogleraar in de Bedrijfskundige Programmatuur van de onderafdeling der Informatica, Technische Hogeschool Twente, en ir. N. J. M. Zwaneveld, informatiesystemen- en automationadviseur, Philips Eindhoven. Ruim 40 vakauteurs hebben bijdragen geleverd voor de inhoud van het nieuwe zakboekje. Het boekje is bestemd voor iedereen die bij wil blijven op het gebied van computes, informatica en automatisering.

De nieuwe uitgave is in de boekhandels verkrijgbaar. De prijs bedraagt f 59,50.

Alle Poly-zakboekjes zijn ook direct te bestellen bij PBNA, Postbus 9053, 6800 GS Arnhem, telefoon 085-716151.

INFORMATION ENGINEERING

An Improved, Automatable Methodology for the Design of Data Sharing Systems.

Ian G. Macdonald

James Martin Associates
11/17 Worple Road
Wimbledon
London SW19 4JS

This paper describes the effect of four years development from the CRIS 1 offering - 'System Development in a Shared Data Environment'. The methodology has been re-developed within James Martin's concept for Information Engineering. It is now comprehensive in its coverage of the development cycle; it is in many respects simpler; it is now designed to be automated and is being supported by workbench software.

1. GENERAL INTRODUCTION

1.1 INTRODUCTION

This paper describes the Information Engineering methodology of James Martin Associates. This methodology has been developed from a concept of James Martin's under the direction of Ian Palmer and the management of Ian Macdonald. In most respects the methodology is in direct evolutionary line from that described in the CRIS 1 paper 'System Development in a Shared Data Environment' by Macdonald and Palmer [6]. Related work can be found in [2, 3, 13, 14, 15]. Key individuals involved with the earlier methodology now work for James Martin Associates. This expertise has however been greatly broadened by others with very different methodology backgrounds and with an international perspective. In addition to this expertise, the availability of real power in personal computers since the beginning of 1984 has allowed a substantial re-alignment of priorities in our approach.

Information Engineering is a graphics oriented discipline. During the analytical stages of a project, business models are created. These are then refined during the design stages to create the technical models that will be used directly for translation into system code. The systems produced therefore have their origins in the business models, so are more likely to serve the business needs than systems produced by a different route.

Information Engineering has been developed with a view to automating its procedures. It is a methodology which is based on a set of tightly integrated techniques which provide information to a common methodology data model. The model is the basis for an active encyclopaedia containing all information relevant to the development process. The techniques are all diagrammatic and provide the primary means of input to the encyclopaedia. The encyclopaedia in turn controls consistency and mapping between techniques. Ultimately the contents of the encyclopaedia drive a system generator. The final result of the integration and automation of Information Engineering is therefore the production of better quality systems and substantial improvements in productivity.

1.2 IFIP CONFERENCING SYSTEM

The analysis presented in the original Macdonald/Palmer paper is not altered substantially by using Information Engineering. The most obvious difference is the addition of action diagrams to represent the logic of procedures and the use of state transition diagrams to represent life cycles of entity types and to represent dialogue flow. However the use of automated techniques eases the task substantially and provides a much greater assurance of completeness and correctness of results.

1.3 ACKNOWLEDGEMENTS

Those most closely associated with these developments have been James Martin, Ian Palmer, John Dodd, Clive Mabey, Leslie Jennison, David Rigby, Shaun Boyle, Kevin Murphy, Arno Oosterhaven, Karel de Graaf, Roger Crane, Roger Tagg, Michael Mills, Keith Short, Bill Dawson, Eric Magnuson, Paul Feldman and Bill Gibson. To all of these we offer sincere thanks. Thanks are also due to Texas Instruments for permission to use their workbench software in illustrating the case study and to our many collaborators in TI who have helped to realise our concepts.

2. THE INFORMATION ENGINEERING METHODOLOGY

2.1 INTRODUCTION

Information Engineering provides a structured framework within which techniques are employed leading to the development of high-quality, integrated information systems. This framework provides a basis for defining and managing projects through the various stages of development.

Figure 1 illustrates the Information Engineering concept in the form of a set of building blocks, each reflecting a technique employed.

The horizontal blocks are the basic techniques used during the stages of the methodology to analyse the business and to specify and construct its systems. The vertical blocks provide strategies for shortening the development life-cycle, given the information gathered beforehand by the basic techniques.

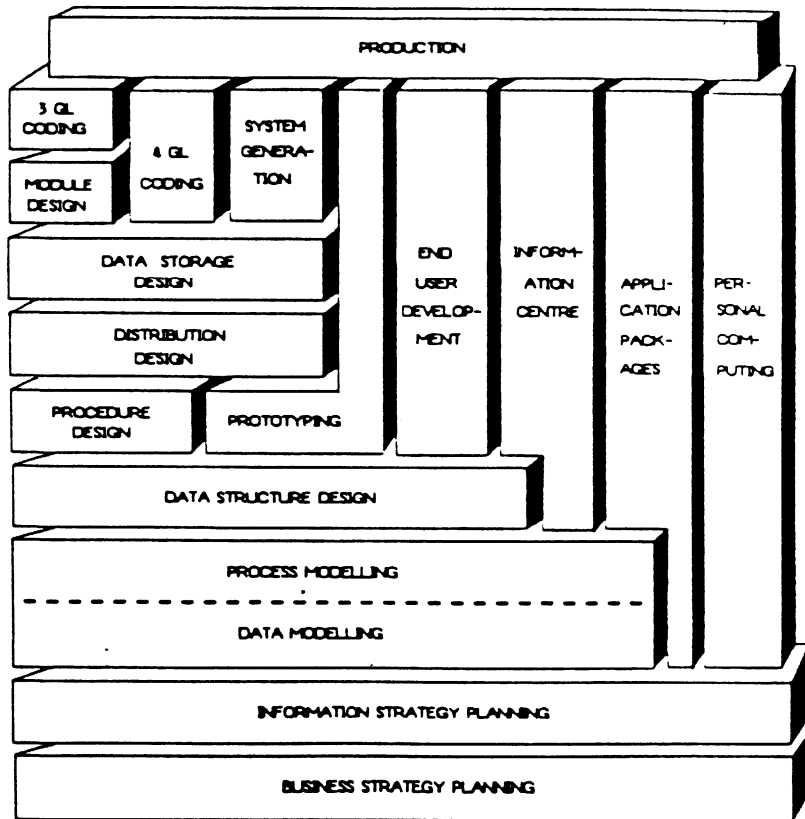


Figure 1: The Major Techniques of Information Engineering

- * The blocks all rest on an enterprise or business model established by Business Strategy Planning. This establishes the mission, objectives and goals of the enterprise in relation to scenarios for the future.
- * The next block is Information Strategy Planning. This is a top-down analysis of the business objectives, their information needs and priorities, the types of data that must be kept, the functions making up the business and how all of these relate to one another. From this, an information architecture is constructed, a business systems architecture is derived and a strategy plan is put together to give direction to subsequent progress. Information Strategy Planning is sometimes carried out across an entire enterprise; sometimes it is carried out for one division, subsidiary or autonomous portion of an enterprise.
- * The third block consists of Data Modelling and Process Modelling. Data modelling is carried out using the entity analysis technique and surveys the types of data needed across an organisation. It creates an entity relationship model which is a broad overview but which does not contain all the details needed for database implementation. Process modelling is carried out using the function analysis technique and breaks down broad functions into specific business processes, identifies interdependencies among them and discovers what data underlies the interdependencies. These analyses provide a comprehensive description or model

of the business without which it is impossible to be successful in implementing comprehensive, enterprise-wide systems.

- * The fourth block is Data Structure Design which creates the preliminary database design and attempts to make it as stable as possible before it is implemented. This provides a mapping from the entity relationship model. It carries no more detail but applies various checks for stability.
- * To create the procedures that employ a database, a diagramming technique (action diagramming) is needed to represent the actions that create data, and retrieve, update, or delete data. The procedure action diagrams are easy to create and are drawn in such a way that they can be converted directly into the code skeletons of fourth generation languages. This is the basis of block 5 - Procedure Design.
- * Analysing how data is used is important for heavy-duty applications. It leads to decisions for distribution design (block 6) and about the physical database design (block 7). For applications with a low transaction volume, analysis of detailed usage is not necessary when appropriate data management systems are used.
- * Finally, Module Design and 3rd Generation Coding can be used to create heavy-duty applications or modules where efficiency is critical and hand-crafting is necessary.

Essential to the entire structure is the set of alternative fast-development paths that enable it to exploit fourth generation technology. These vertical paths show when each of today's strategic techniques can be used. They show the earliest point at which it is reasonable to break out of the development cycle and proceed directly to production systems. This combination of solid building blocks and fourth generation techniques satisfies our two major objectives - to provide better quality systems and to achieve greater productivity in building those systems.

2.2 INFORMATION ENGINEERING STAGES

2.2.1 INTRODUCTION

Information Engineering projects are organised around the stage framework shown in Figure 2. Being task oriented this figure is a useful means of bringing out the practical aspects of the methodology. It is also sufficiently general to enable the place of the techniques and tools to be seen clearly.

2.2.2 BUSINESS STRATEGY PLANNING

This is not formally part of the methodology, since it is not normally carried out by staff in the information systems area, but it is recognised as a necessary precursor.

The end product of Business Strategy Planning will be a Business Plan, indicating overall business goals and strategies. This should also show the main business functions and organisational structure, and the objectives established for each function. It must be framed in sufficiently quantitative terms for business information needs and priorities to be inferred.

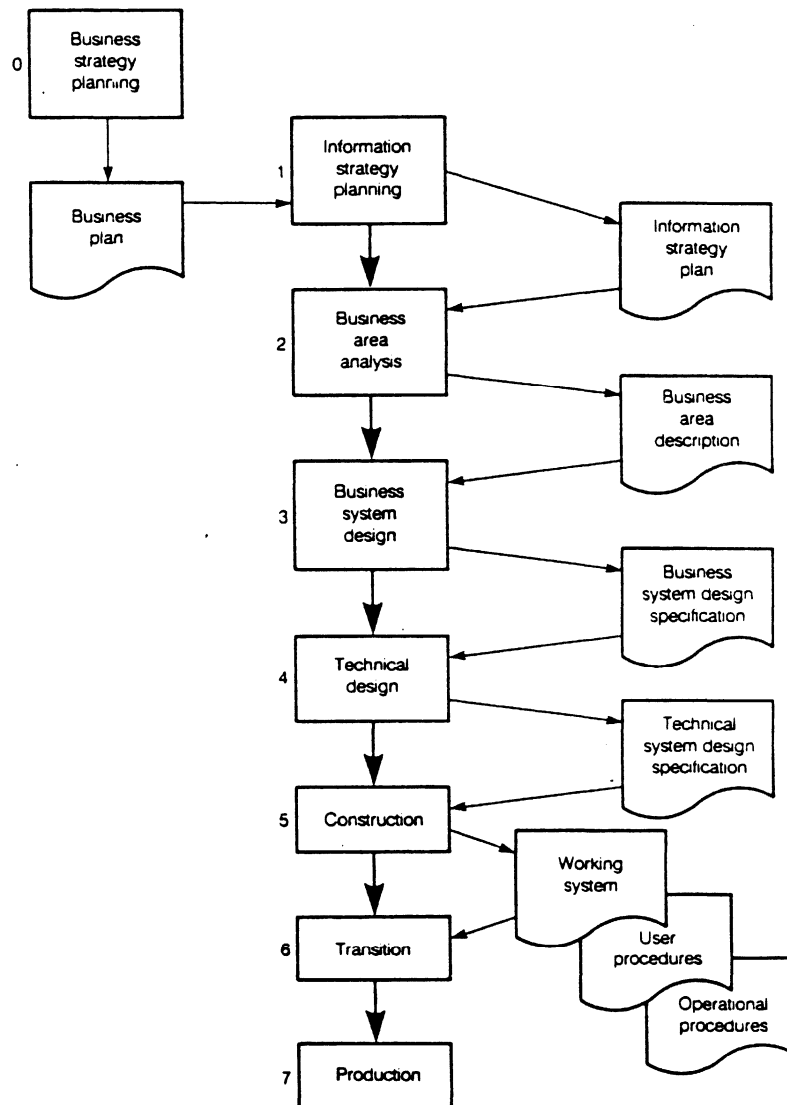


Figure 2: The Stage Framework

2.2.3 INFORMATION STRATEGY PLANNING

An overview is taken of part or all of the enterprise in terms of its business objectives and related information needs, its principal types of data (subject areas) and its business functions. This leads to the construction of an information architecture expressed in terms of a data model (entity relationship model) and a function model (decomposition and dependency models). The architecture is ultimately divided into a number of business areas, each having the scope of a possible analysis project.

The strengths and weaknesses of the principal current systems of the enterprise are also assessed. The systems structure is modelled so that a migration path can be devised which provides a clear route towards the ideal structure of the architecture. This aims at protecting the present investment in systems wherever it remains effective.

Based on these analyses a business systems architecture is drawn up and priorities for information systems development are set. A systems development strategy is produced which expresses priorities for addressing the systems needs of all business areas.

Information Strategy Planning results in an Information Strategy Plan. This includes the information architecture and the business systems architecture. It also includes:

- a technical architecture which provides a statement of direction for the enterprise's computer hardware, software and communications facilities;
- a proposal for the organisation of the information systems function, to satisfy the demands of the strategy;
- a broad business evaluation;
- a migration path;
- a plan for systems development, including work programs for high priority projects.

Once the plan is complete, remaining issues should not be material to the strategic direction, i.e. are unlikely to change the strategic plans. Results should be reasonably insensitive to the perceivable range of changes in business and technical plans, volumes and costs.

This approach enables the enterprise to establish the best possible underlying systems and database architecture from which its developers can design and implement a coherent set of information systems. The strategy also establishes a context for re-evaluating assumptions and priorities so that it can be controlled, managed and revised at regular intervals.

A typical task structure for Information Strategy Planning is illustrated in figure 3.

- * Current Situation Analysis leads quickly to an overview of the enterprise. The business strategy is clarified through discussion with top management; the organisation, technology and preliminary information architecture (data subject areas and business functions) are explored through existing documentation.
- * Executive Requirements Analysis provides a way for managers to describe their objectives, information needs, activities, priorities and perceptions of the current systems - from which analysts can prepare architectures.

Current Situation Analysis

- Analyse the business strategy
- Analyse the information system organisation
- Analyse the technical environment
- Define the preliminary information architecture

Executive Requirements Analysis

- Determine information needs
- Determine priorities, responsibilities, problems.

Architecture Definition

- Define information architecture
- Analyse distribution
- Define business systems architecture
- Define technical architecture
- Define IS organisation

Information Strategy Plan Creation

- Determine business areas
- Prepare business evaluation
- Prepare information strategy plan

Figure 3: Information Strategy Planning Tasks

- * Architecture Definition is:
 - * first the preparation of the overall information architecture, based on entity types within the subject areas and on the decomposition of functions of the business;
 - * then the derivation from the information architecture of a business systems architecture showing an ideal view of the systems required at all levels in the enterprise;
 - * then the development of a view of the technology required in support of the systems;

- * and finally the proposing of an organisation for the information systems function which will enable it to give effective support to the development and operation of the systems and their technical environment.
- * Information Strategy Plan Creation segments the information architecture into business areas each with the scope of a potential analysis project. It then evaluates strategies for achieving the architectures, including migration from the current systems, and prepares a prioritised strategy plan for user agreement.

2.2.4 BUSINESS AREA ANALYSIS

For an identified business area within the scope of the Information Strategy Plan, a detailed study is carried out of its data, activities and all interactions between the two. This leads to identification of entity types and elementary business processes and the input and output information views of the processes. These are analysed in detail and their names, interactions, meanings, quantities, conditions and business algorithms documented. An important feature is that maximum involvement of end users in the specification of requirements, priorities and resulting systems facilities is recommended.

At the end of Business Area Analysis a Business Area Description is prepared showing its business functions, the processes making up the functions and the process dependencies, also the area's entity types, relationships and attributes, with their properties and their usage patterns in the business processes. These provide much greater detail than the information architecture and indicate information needs and priorities within the area.

From this information, a detailed statement of the business requirement for information systems in the area is produced.

It is then possible to identify the broad nature of likely computer support required for business processes and to define the scope of one or more business systems to be designed (design areas) and a work program and resource estimates for them. All the information is present (about the business and its user's requirements) which is necessary to select particular business processes for computer support and to design the computer systems and the data structures needed to give that support.

The task structure for Business Area Analysis is illustrated in figure 4. An important point to note is that data and processes are analysed in parallel although the emphasis is on defining the data structure correctly as a guide to the proper definition of processes.

- * Entity and Function Analysis are the major tasks of this stage. They are used to gather information on all elements of the business area. They expand the information architecture to form detailed data and process models which are the foundation of all the resulting information systems.

Entity and Function Analysis

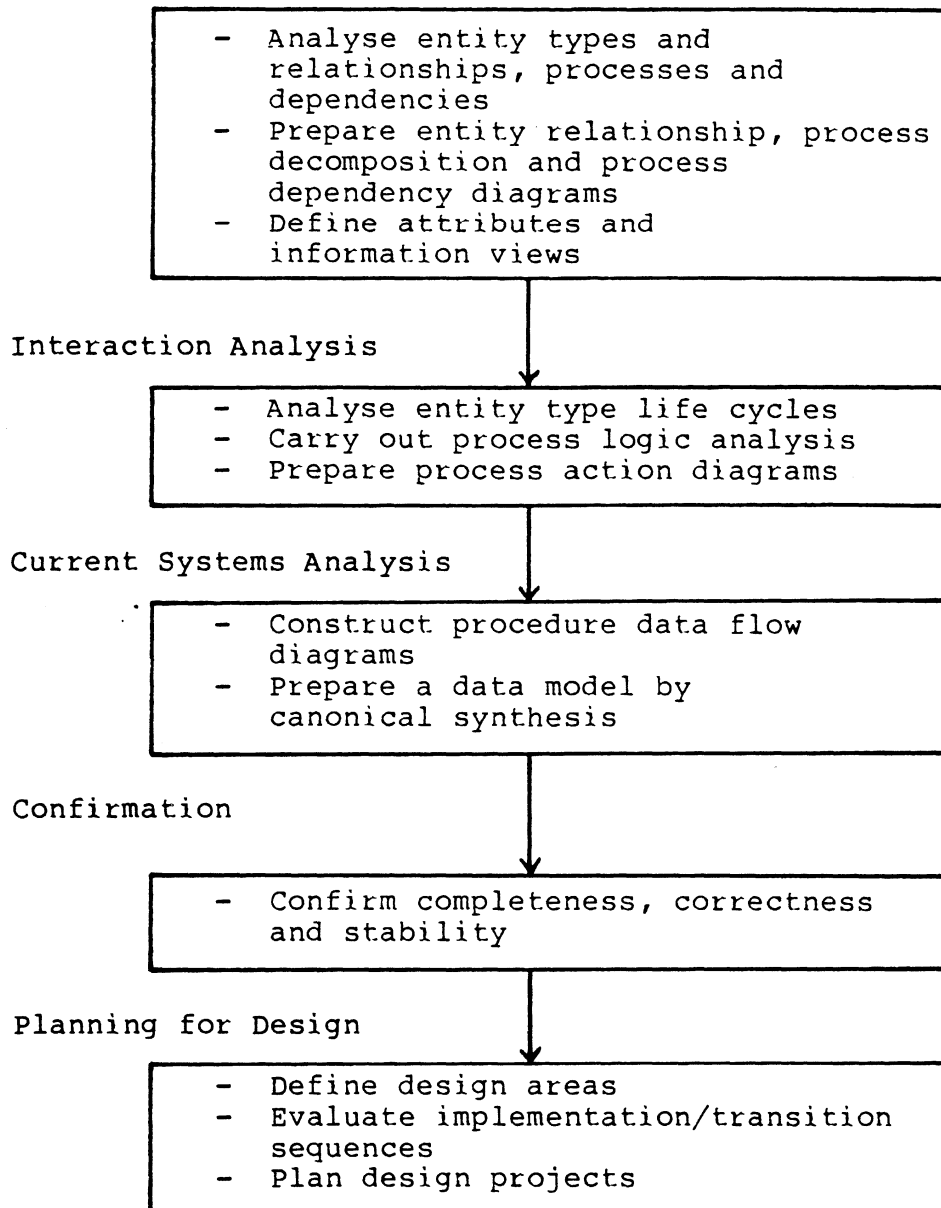


Figure 4: Business Area Analysis Tasks

- * Interaction Analysis looks at the data used by each process, and conversely the processes acting on each entity type. It provides a view of the business dynamics and a cross-check on the models.
- * Current Systems Analysis creates models of the systems existing or under development. This is done using the same conventions as those in entity and function analysis so that the models can be compared for confirmation and transition analysis purposes.

- * Confirmation formally checks out all of the results.
- * Planning for Design selects those parts of the models to be automated and works out a plan for the design sequence.

2.2.5 BUSINESS SYSTEM DESIGN

For the whole or a major part of a business area analysed (a design area), the facts gathered during analysis are used to design a system to meet the identified business requirements. The design includes all those parts of the system directly relevant to its users including procedures (to which the business processes are mapped), dialogues and controls. It is kept as independent as possible of the technology to be employed in implementation. Prototyping techniques are used to replace many of the tasks traditionally gone through in this stage.

An important objective of this stage is that it should complete the system design to the extent possible without pre-judging technical issues. It is also heavily user oriented and requires agreement by these users on the ways in which they will interact with the system.

The final product from Business System Design is a Business System Specification showing, for each business process, the consolidated documentation of information flows and user procedures and, for each computer procedure, a consolidated and confirmed version of the results of business area analysis, plus the dialogue design, screens, reports and other user interfaces, an action diagram for the procedure and adjustments to the data usage patterns. From this, a detailed scoping and data flow diagram of the intended computer systems is prepared together with a work program and resource estimates for the next stage.

Once this is done, all aspects of the system which relate directly to the user should be defined and stable, and sufficient information should exist to finalise estimates for, and to complete, technical design.

A typical task sequence for Business System Design is illustrated in figure 5. This is varied depending on whether prototyping, system generation or conventional construction approaches are to be used.

- * Preliminary Data Structure Design applies the rules of the chosen data management system to the entity relationship model for a business area to obtain a first data structure. Note that the scope of this task is greater than that of the other Business System Design tasks (the whole business area, not just the design area). This guarantees effective integration and compatibility for all systems supporting the business area.
- * System Structure Design maps the business processes of the design area to procedures and shows how they interact within the system, by creating a data flow diagram.

Preliminary Data Structure Design

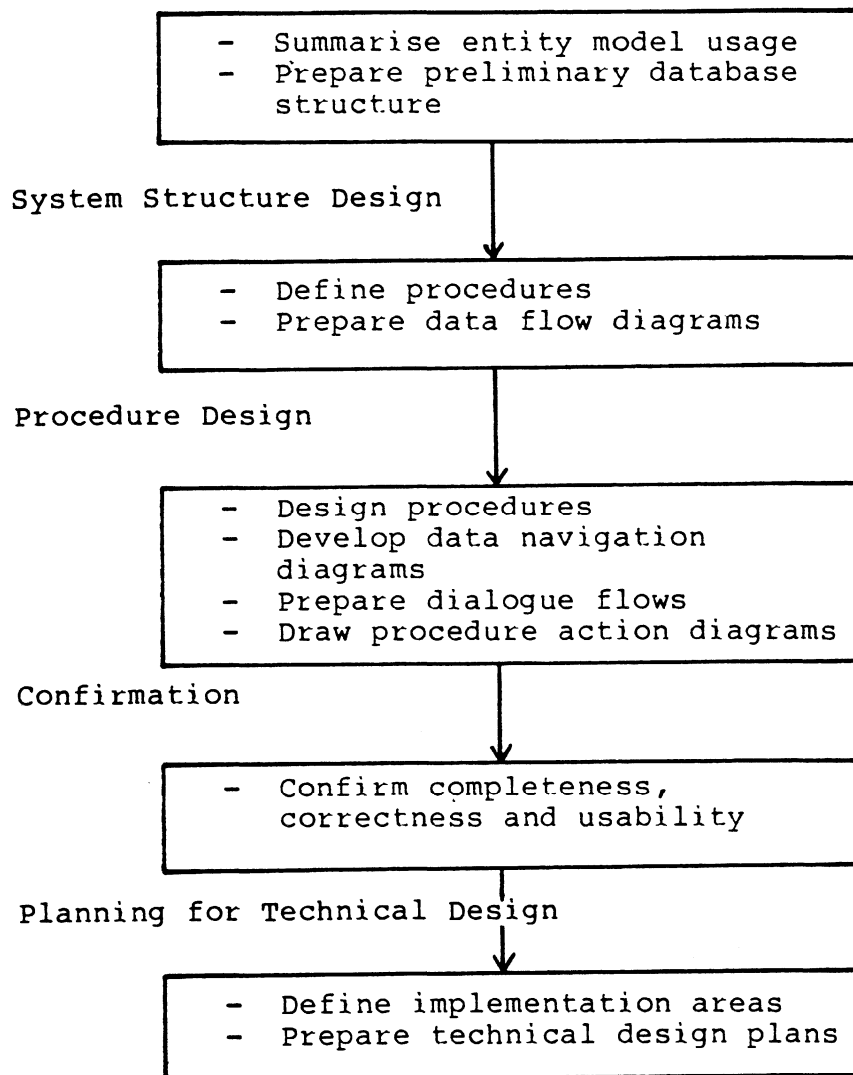


Figure 5: Business System Design Tasks

- * Procedure Design develops dialogues and input/output layouts for each procedure. It provides the logic of the procedure, including control and error handling, in the structured form given by procedure action diagrams. This can readily be done by prototyping and can lead directly to the generation of code from the action diagrams.
- * Confirmation checks all the results.
- * Planning for Technical Design defines what is to be done in the next stage, including the possible subdivision of the system into smaller implementation units.

2.2.6 TECHNICAL DESIGN

For the computerised aspects of the business system specified above, the facts gathered during analysis are used to design those parts of the system which are dependent upon the computer technical environment. This is carried out in sufficient detail

Data Design

- Prepare database load matrices
- Refine database structure
- Design data storage
- Design other files

Software Design

- Define programs, modules and integration groups
- Design programs/modules
- Define test conditions and cycles

Transition Design

- Design software and procedures for bridging and conversion
- Plan system fanout
- Define user training

Operations Design

- Design security/contingency procedures
- Design operating and performance monitoring procedures
- Design software for operations

Verification of Design

- Run benchmark tests
- Assess performance

System Test Design

- Define system tests
- Define acceptance tests

Implementation Planning

- Reassess costs
- Prepare implementation plan

Figure 6: Technical Design Tasks

for construction and operation to be adequately costed. This design includes logical and physical data structures, computer programs, operational procedures and interfaces. The level of detail in the design is dependent upon the selected implementation vehicle. Certain vehicles, e.g. system generators, have much of the technical requirement pre-defined.

The aims during this stage are to define efficient computer systems to support the selected business processes and to develop good (plus or minus twenty percent) estimates of costs and timescales for construction and transition, in terms of manpower and computer equipment.

The end result of Technical Design is therefore a Technical Specification containing database designs and the application system technical design including batch runs, finalised conversation flows and definition of programming work units. It also includes the technical standards for the system - the hardware and software environment selected, its mode of use and specific standards and conventions proposed.

Finally it identifies the content of the construction and transition stages and gives a work program and resource estimates for these stages.

This Technical Specification should provide a stable design which meets functional and performance objectives and is insensitive to predicted business and technical changes.

A typical task structure for Technical Design carried out manually is illustrated in figure 6.

In practice fourth generation tools such as system generators are enabling much of this stage to be automated. Only the design of operational procedures is not well supported although the use of tools introduces an element of standardisation to them.

2.2.7 CONSTRUCTION

For each implementation unit identified during design, a system is put together. This includes installation of equipment, establishing files, setting up procedures and specifying, coding and testing programs. The aim in Construction is to develop an application system as defined in the technical specification which is on target as to timescale and budget, of an acceptable quality and which contains all necessary operating and user procedures.

The stage can be regarded as complete once the defined acceptance criteria for the application are met satisfactorily, covering: systems functionality, stress testing, operational procedures, user interfaces.

The sequence of Construction tasks is shown in figure 7.

System Generation

- Construct computing environment
- Prepare development procedures
- Construct database/files
- Generate modules
- Generate module test and execution data
- Perform integration tests
- Generate documents

System Verification

- Generate system test and execution data
- Perform system tests
- Generate acceptance test and execution data
- Perform acceptance tests
- Gain approval

Figure 7: Construction Tasks

A major aim in Information Engineering is to automate the Construction stage to as great an extent as is possible. The objective in system generation is to produce provably correct code. Testing then becomes unnecessary and all that is required is a demonstration to confirm that the system generated does perform as the user expected. If not, alterations to the system are accomplished by code re-generation from an amended specification.

2.2.8 TRANSITION

Transition is the phased replacement of existing procedures and files by the new system and data stores. It is governed by the Transition Plan, including a work program and resource estimates, which is normally finalised in parallel with the Construction stage, although it is not really dependent on the outcome.

Transition can be regarded as successful when the system operates for a specified period within defined tolerances as regards performance, error rate and usability, and passes its post-implementation review.

The Transition tasks are illustrated in figure 8.

Preparation

- Prepare transition schedule
- Train users
- Install new local hardware

Installation of New Software

- Perform conversion
- Execute trial run

Final Acceptance

- Agree terms
- Move fully to new system
- Prepare regression tests

Fanout

- Prepare a location
- Install at a location

System Variant Development

- Identify requirement
- Revise analysis and design
- Perform construction and transition

Figure 8: Transition Tasks

2.2.9 PRODUCTION

Production is the successful operation of the system, with tuning and modification as necessary, until eventually the transition stage in some other project replaces the systems built in this project.

The main objectives in Production are to maintain service levels and functional performance during the lifetime of the system and to respond promptly and effectively to changes in business requirements.

The Production tasks are illustrated in figure 9.

Evaluate system

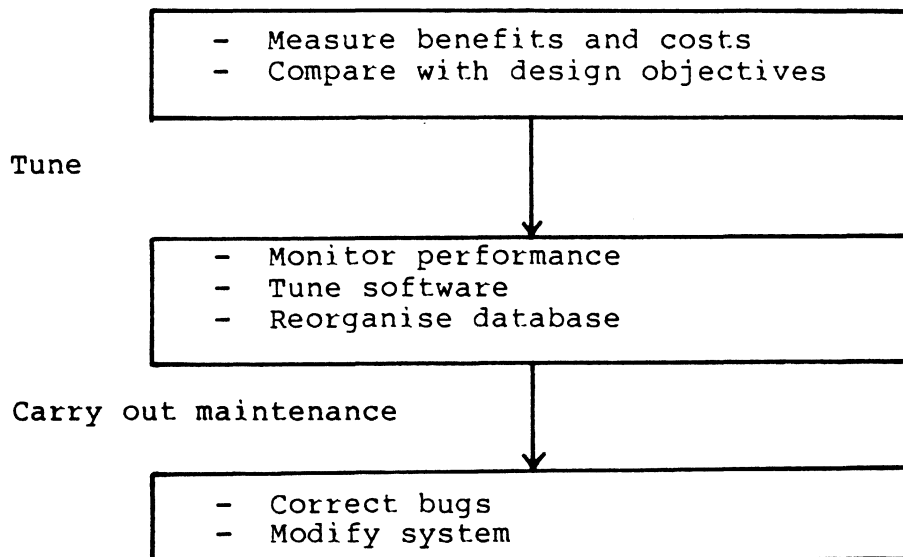


Figure 9: Production Tasks

Any changes to the system required during its life can be accomplished by system re-generation, if a systems generator has been employed.

2.2.10 CONTROL

It is an Information Engineering objective that all tasks performed in all stages of the methodology should yield outputs which can be measured on a quantitative basis. Further, the size of each measurable work unit should be fairly small (about 1-10 man days) to avoid waste of resources on unproductive tasks. This is the key factor for productivity measurement and control. To this end, work programs and estimating guidelines are provided and productivity is assessed by function point measurement [1].

2.3 ADDITIONAL TECHNIQUES

Information Engineering contains a number of techniques which were not presented in the earlier Macdonald/Palmer paper. Several of these support Information Strategy Planning. These will not be discussed since they are outside the scope of the IFIP case study. Others support analysis and design. They are:

- * data analysis diagramming
- * state transition diagramming
- * data flow diagramming
- * data structure diagramming
- * action diagramming

2.3.1 DATA ANALYSIS DIAGRAMMING

Data analysis diagrams, also known as bubble charts, show data items and the associations (functional dependencies) between them, as seen in figure 16. They are described in detail by Martin [9].

Each data item is represented as an ellipse, and each association as a line between two ellipses. Cardinality and the direction of functional dependency are indicated by the correct placement of crow's foot and bar symbols.

These diagrams enable us to identify groups of data items that are functionally dependent upon a single data item which is termed the primary key. These data item groups may be considered as record types or candidate entity types.

Data analysis diagrams are used in the Business Area Analysis stage to analyse fields in current systems or fields required in business processes. They are then used in Business System Design to confirm that designed user views are compatible with the business's data model. They also provide a pictorial way of representing users' views of layouts: eg. the structure of reports, screens and input forms (either existing ones or proposed ones). The ellipses then represent the fields of these layouts.

2.3.2 STATE TRANSITION DIAGRAMMING

State transition diagrams are used to document details of the life cycles of things such as entity types and dialogue screens which are changed by the activities with which they are involved.

For example, a person may be an Applicant, an Employee, or a Pensioner during his time with a company. He can become involved in different business processes depending on which of these states he is in. Some of these processes can change the state (eg. Hire an Applicant) and this is recorded as a state transition.

The possible states in which a thing can exist are shown as horizontal lines. The transitions among these states are shown as vertical lines. The event or procedure causing each transition may be indicated on the diagram in conjunction with the vertical lines.

This technique is used in Business Area Analysis for entity life cycle analysis, and in Business System Design to show interactions between various screens and the procedures which take place to support the screens.

Figures 19 illustrates its use for life cycle diagramming

2.3.3 DATA FLOW DIAGRAMMING

Data flow diagrams are used to illustrate associations between designed activities - procedures, program modules or systems. They are therefore very similar to dependency diagrams, the difference being that they show the actual flows of data among procedures. Figure 24 illustrates a typical example.

Data flow diagrams can be used by strategy planners and analysts to show the overall architecture, or individual structures, of current systems. They are most commonly used by designers as a relatively high-level design tool for mapping out the procedures and data stores required to automate a given design area.

Data flow diagrams show the flows of data between each procedure in the design area. They also show flows of data between procedures and data stores and flows across the boundaries of the design area.

Like other major diagram types, notation can be added to show optionality, conditions, mutual exclusivity, one-to-many association, sequence and parallelism. Separate diagrams may be drawn to represent different levels of decomposition.

2.3.4 DATA STRUCTURE DIAGRAMMING

Data structure diagrams show the structure of data used in database or file systems. They illustrate the record types included in the design, the types of linkages between those record types and any direct entry mechanisms (indexes or algorithms) used in acquiring specific records.

The diagram uses rectangular boxes to represent record types, lines between the boxes to represent linkage types and banners (elongated triangles) to represent entry points. Other standard forms of notation can be added to the diagram. Figure 25 illustrates a typical diagram.

These diagrams are used by designers to prepare the preliminary data structure during Business System Design and the refined data structure during Technical Design.

2.3.5 ACTION DIAGRAMMING

An action diagram expresses the logic of a process, procedure or program. It consists of a list of actions, and brackets which indicate whether groups of actions are sequential, conditional or repeated. These are the standard structured programming constructs. In addition, action diagrams enable us to represent concurrent actions.

Further notations permit us to show escape actions and to show where sub-procedures are executed. Inputs and outputs can also be added to action diagrams enabling automated tools to correlate and validate sets of action diagrams.

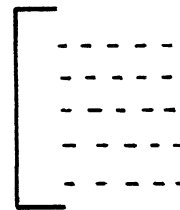
Action diagrams are used in Business Area Analysis, where process action diagrams describe the logic of processes; in Business System Design, where procedure action diagrams define the logic of procedures and dialogue steps; in Technical Design, where program action diagrams define the logic of programs and modules.

The syntax of the action diagram statements can be varied to suit the programming language to be used (except for process action diagrams). When a procedural fourth generation language is employed the procedure action diagrams can be converted directly into program statements. So action diagrams facilitate a step-at-a-time decomposition from high-level design down to executable code.

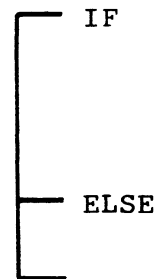
Figure 22 illustrates a simple action diagram.

The constructs of action diagramming are as follows:

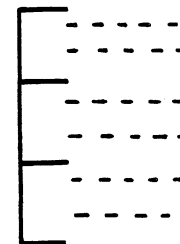
A simple bracket groups a set of actions which are executed once, in the sequence they appear in.



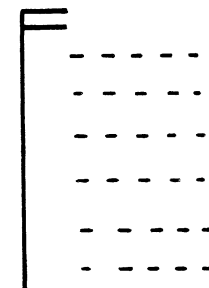
An IF statement added to a simple bracket implies that the execution of the set of actions is conditional. An ELSE provides an alternative.



A divided bracket provides a case structure in which the groups of actions are mutually exclusive.



A double line heading on a bracket represents repetition. The group of actions is executed multiple times.



Nesting of brackets expresses hierarchical subdivision of the logic of the overall set of actions.



An arrow passing back through a bracket represents an Escape. Control then transfers to the left-most bracket to which the arrow leads.



An arrow within a repetition bracket forces the next iteration.



A semicircle connecting two brackets implies the possibility of concurrent execution of the two sets of actions.



3. ADVANCES IN THE METHODOLOGY

3.1 SCOPE

3.1.1 SUPPORT

To be complete in a practical sense a methodology has to be more than a good set of ideas, however clearly expressed. The earlier Macdonald/Palmer paper described a methodology which had a clearly defined framework and a number of well developed techniques. A good body of training material existed for the analysis stage and for some aspects of design. No other substantial documentation existed, other than papers in journals. To a significant extent, any use of the methodology depended on support from experienced consultants.

Information Engineering is much more accessible. The concepts are available in books by Martin [7-11]. The methodology is fully documented in nine detailed reference guides and in a number of introductory publications. Training is available at three levels of detail and covers the information strategy planning, business area analysis, business system design and technical design stages. It also includes explanatory material for presenters of the courses. Interactive video disc versions of courses are being prepared to remove the need for formal lecturing. Detailed case studies illustrate the use of the methodology and software supports many of its techniques. Detailing the methodology in this manner de-mystifies it and means that it can be used with lower levels of specialist support [12].

3.1.2 STAGES

The two major benefits from Information Engineering are improved quality of systems and improved productivity in development. The first is achieved by focussing more clearly on systems which will satisfy business objectives; by tightly integrating the techniques used in development; by automating the development process. The second is achieved by breaking free from the traditional development cycle and adopting strategies to shorten it (see section 2.1 and figure 1) and again by automating wherever possible. Some types of system, however, continue to require a hand-crafted approach, so the methodology defines a life-cycle and stages around which projects can be set up and controlled. The main differences from the CRIS 1 paper are as follows:

* Information Strategy Planning

This replaces the 'strategy stage'. The need for a sound architecture on which to plan an enterprise's development of systems is now much more clearly recognised. This must be closely tied to the enterprise's own business plan and therefore to business objectives, goals and critical success factors and the information needs which flow from them.

Information Strategy Planning prepares an information architecture which is the combination of a global entity relationship model and a function model for the enterprise. From this a business systems architecture is developed to satisfy information needs. In parallel, a technical architecture is derived to show the kind of hardware/software/communications environment needed to support the business systems. The information systems organisation structure is also evaluated to provide an organisation which will allow the architectures to be implemented. Finally projects are scoped out and prioritised and an information strategy plan created to guide subsequent development.

* Business Area Analysis

This replaces the 'analysis stage'. Broadly speaking this stage is the same. Diagramming conventions have been rationalised and techniques for interaction analysis have been improved. Process dependency diagrams are more rigorously specified and demand clear definition of the views of information both being made available and used by the business processes. They therefore demonstrate that the interdependence between processes is fully understood. Entity type life cycle analysis is now done using a simpler and more rigorous state transition diagram. Action diagrams have been introduced to provide a clear presentation of the logic of activities at any level. They therefore replace the cumbersome 'function logic specification form' and can be mapped to from state transition diagrams and dependency diagrams.

The analysis of current systems for transition purposes is greatly enhanced. Bubble charting and canonical synthesis [9] is used to model existing data stores in a manner that can then be matched with the entity relationship model. Data flow diagramming is used to model procedures within the system in a manner that can then be compared with the process dependency diagram. Significant effort is also put into verifying the completeness and correctness of models and this is extended to a thorough review of their stability in the face of hypotheses about conceivable changes to the business.

* Business System Design

The 'design stage' is broken in two - Business System Design which is concerned with user-oriented, behavioural aspects of the system and Technical Design which is concerned with achieving efficiency in a defined hardware/software environment.

Business System Design maps an entity relationship model to a preliminary data(base) structure. It maps business processes to implementable procedures and uses data flow diagramming to define their associations within the system. Each dialogue is defined using a form of state transition diagram and finally each procedure is represented in action diagram form and can then, readily be translated into any one of a number of 'fourth generation languages'.

* Technical Design and Construction

It is an aim of Information Engineering that these stages should be replaced by the use of system generators and fourth-generation languages. The type of system being developed will determine the type of tool most appropriate for the implementation.

* Transition

The importance of transition requirements and of the transition effort is now heavily stressed and dealt with in detail at every stage. The value of the current investment in systems is recognised and protected by focussing on the need for co-existence, bridging or conversion of systems where appropriate.

* Project Management

This continues to receive heavy emphasis and is enhanced by the addition of the function point analysis technique [1] for measuring productivity improvement and for assessing the scale of projects.

3.2 NOMENCLATURE

Some naming conventions have altered. Figure 10 illustrates the structure of the major objects in the methodology. There is nothing novel about the names used but this diagram does emphasise the parallelism we stress between data and activities.

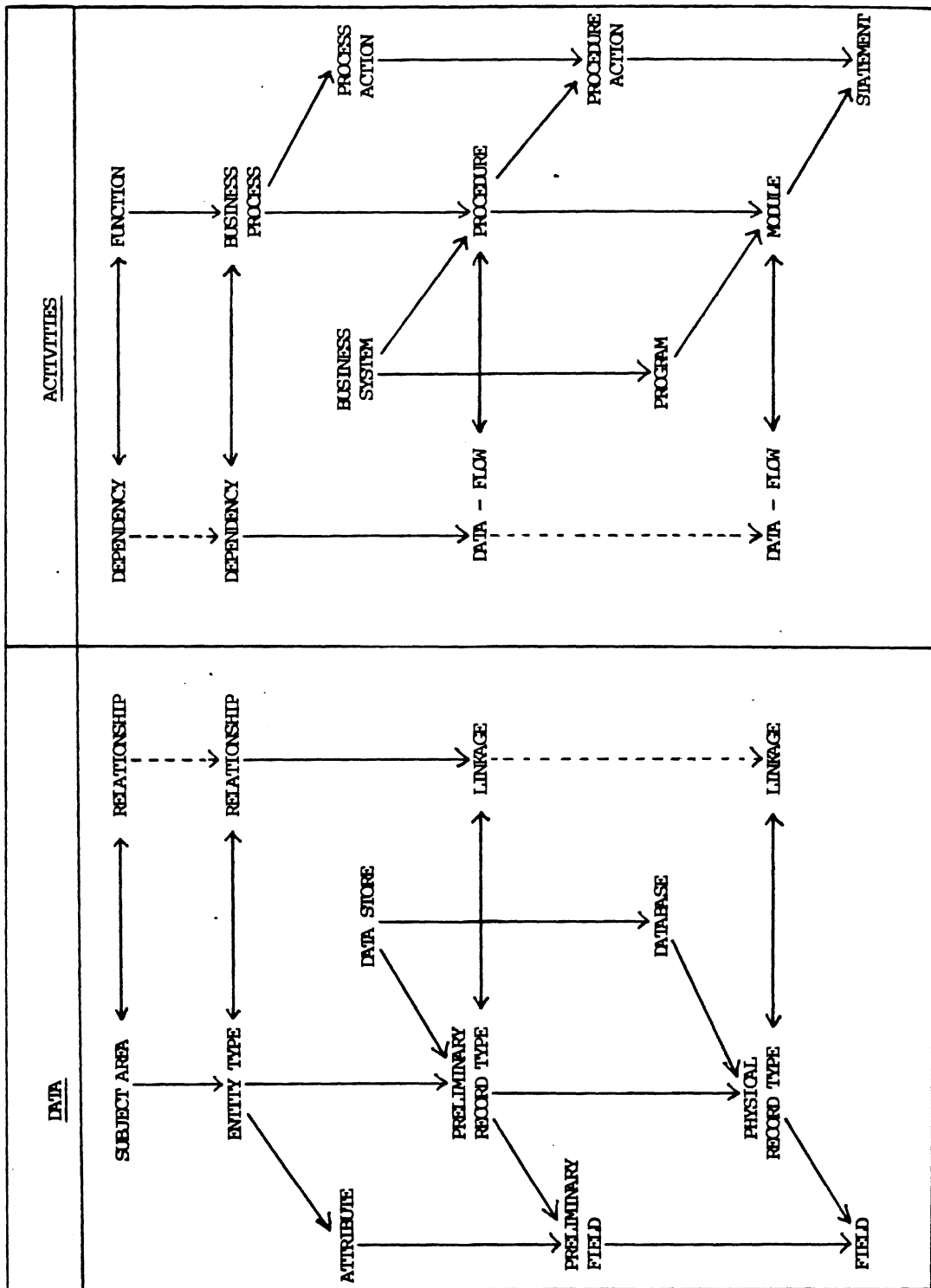


Figure 10. Major Object Types in the Methodology

3.3 DIAGRAMMING

Diagrams are an important means of simplifying communication among those involved in Information Engineering projects, whether end users or methodology specialists. Each technique is oriented towards diagramming and a diagram is delivered by every major task in the methodology.

Diagrams can be a most rigorous form of representation particularly if it can be demonstrated that related diagram types can be transformed or mapped from one to the other without introducing inconsistencies. Accurate, controllable progression through the methodology is then assured and a basis for automation exists, with the encyclopaedia providing the information and mappings for each form of diagrammatic representation.

To achieve rigour in the diagrams and simplify their automation two features have been stressed:

- (1) the parallelism between ways of representing data and activities.

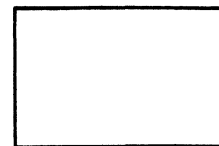
Diagrams with similar purpose should have a similar style. A hierarchical decomposition of data objects should look like a decomposition of activities since the meaning of decomposition is always the same. Likewise a diagram to represent associations between data objects (eg. an entity relationship diagram) should bear a strong resemblance with a diagram to represent associations between activities (eg. a process dependency diagram).

- (2) consistency of meaning in symbols.

Traditionally, methodologies have tended to stress differences between forms of representation of system objects. Information Engineering stresses similarities. A data object is always a rectangle irrespective of the stage in the methodology. An entity type therefore has the same visual significance as the record type to which it has been mapped so the viewer of the diagram need not be concerned with the mapping rules. Likewise a cardinality of many is always represented by a crow's foot irrespective of the type of diagram.

An important effect of these points is that it reduces the number of symbols needed for the whole set of diagrams. The principal symbols are as follows:

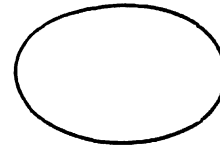
A rectangle represents a type of data
eg. a subject area, entity type, record
type or data store.



A soft box represents a type of
activity
eg. a function, process, procedure,
module or program.



An ellipse represents a component of a data object
eg. an attribute or field.



A square bracket represents a group of components of an activity type
eg. a block of actions or statements.



A bar right across a diagram represents a state in the life of some object
eg. an entity state or dialogue step.



A plain line represents an association between data objects
eg. a relationship or linkage.



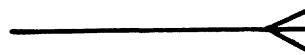
A directed line represents an association between types of activity
eg. a dependency or data flow.



A small bar across a line represents a cardinality of one.



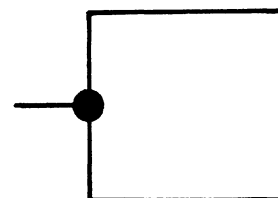
A crow's foot on a line represents a cardinality of many.



An 0 on the line represents a cardinality of zero (or optionality).

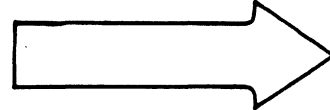


A solid circle at an intersection of lines represents mutual exclusivity among the associations leading away.

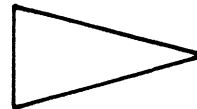


Note. Circles (solid or open) are always associated with a condition. The condition will cover either existence or execution.

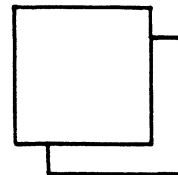
An open arrow represents an event which triggers an activity.



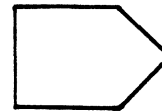
A triangle represents an entry point into a designed data structure.



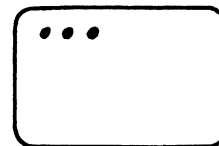
A double square represents a terminator - an activity beyond the bounds of a system or project from or to which inputs and outputs are directed.



A pentagon represents a connector within a diagram when it is especially large or complex.



An ellipsis shows that an aspect of, or object in, a diagram can be exploded to show more detail.



3.4 AUTOMATION

3.4.1 KEY FEATURES

Key requirements of an architecture for automation and a set of tools can be summarised as follows:

- * The tools must support the capture of all information relating to the business and technical environment. This information describes data, processes and their interactions, and includes the designs for interactions with users.
- * The language of representation must use a common syntax across all the intended tools and through all the stages of the methodology. Above all it should be easy to use and, wherever possible, graphic in nature.
- * The internal representation of the models must be sufficiently rigorous to allow the mapping of information from each stage of the methodology to the next. The lowest

level mapping from design specification to program code must be automatic, i.e. there should be automatic code generation.

- * The tools must be able to perform complex analyses and queries on the stored information. These should include:
 - automatic diagram synthesis from stored models.
 - clustering algorithms applied to matrices
 - distribution analysis, e.g. algorithms to specify locations for databases or processors
 - impact analysis, including "where used" type queries
 - performance simulation for prototype applications
 - abstraction and reverse engineering of summary
 - information from detailed information
- * All the tools should act upon a single repository of information describing the business and its information systems. The information should be stored in a non-graphical form and be used in generating consistent sets of diagrams.

This repository should provide the basis for integrating all views of information, automatically checking consistency and maintaining mappings between levels of information.

- * Adequate access and update authorisation control should exist, with the ability to restrict workstations to subsets of the global model.
- * Version control should exist for all forms of work in progress.
- * Status reporting and query facilities should be provided to allow progress monitoring on projects.
- * The tools should, wherever possible, be available on distributed workstations, particularly portable equipment to allow easier user participation in the analysis and design.

3.4.2 THE INFORMATION ENGINEER

The need for automated development has long been recognised and has been clearly described by Martin [10]. Our approach to satisfying the need has been captured in the form of a concept for the architecture of a product - the Information Engineer.

The Information Engineer consists of several integrated sub-architectures. The interaction of these sub-architectures to form the global Information Engineer architecture can be visualised as in Figure 11. The central encyclopaedia architecture interfaces to each of the dependent processing architectures and all are enveloped by the Environment Control architecture.

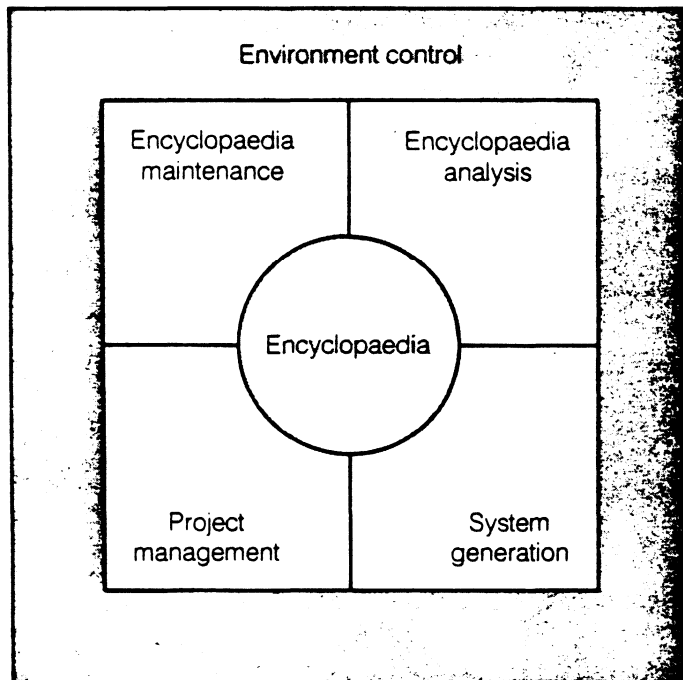


Figure 11: Global Architecture for the Information Engineer

* The Encyclopaedia

A central concept in the Information Engineering methodology, and in any attempt to automate it, is the unbroken flow of information from one stage in the life cycle to the next. In real terms this information flow is in the form of an extracted subset of an earlier model, chosen for detailed study at the later stage. No actual flow of data need occur, rather existing definitions can be enhanced in-situ. This requires a repository which can accept information of varying degrees of detail whilst maintaining a full version history. The information includes details of all forms of data and activities; of associations among each; of conditions governing their existence or behaviour; of the usage of information; and of items describing the technical environment and items for project management purposes.

The encyclopaedia implements the entity relationship model for the methodology. This meta data architecture must exist before the other architectures are defined. Detailed work on this architecture has shown a requirement for over one hundred different object types with several hundred relationships existing among them. This is too elaborate to reproduce here, but figure 12 illustrates a generalised view of the structure needed to support business area analysis.

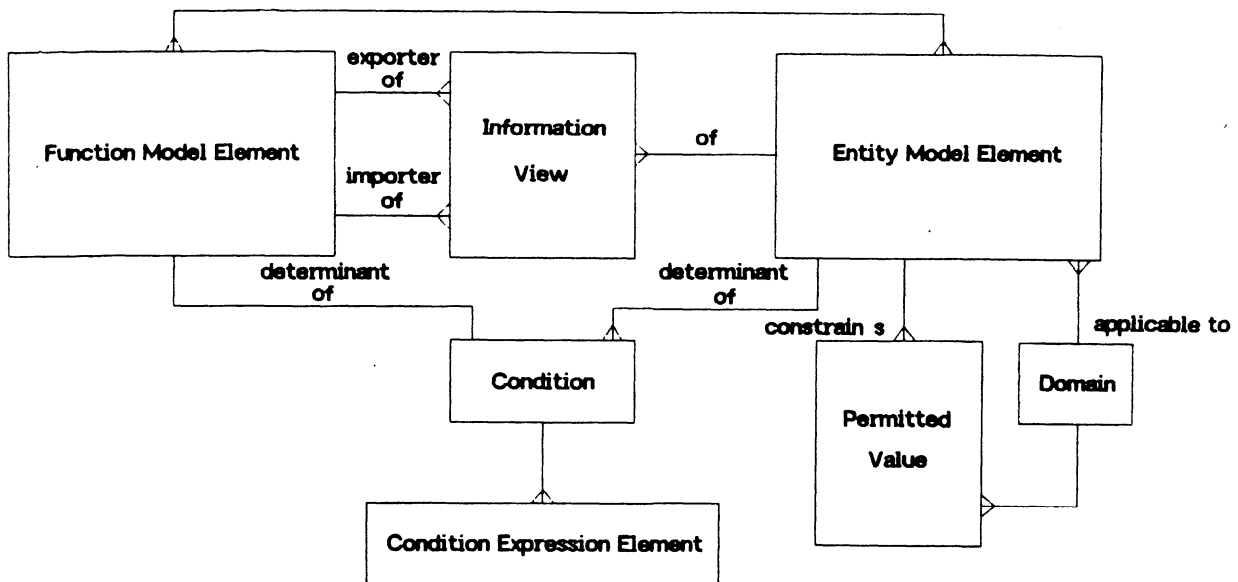


Figure 12: BAA Meta-Model Overview

* Encyclopaedia Maintenance

Encyclopaedia Maintenance implements the process architecture describing a minimum set of structured graphics tools and matrix handling tools that will allow the creation and maintenance of all the encyclopaedia object types and their associated properties. This architecture includes the 'prompter', which guides the analyst or designer through his tasks.

* Encyclopaedia Analysis

Encyclopaedia Analysis implements the process architecture describing a set of facilities to allow complex algorithm based analyses and searches to be performed on the encyclopaedia or on matrix held subsets. Facilities include Impact Analysis, Cluster Analysis, Canonical Synthesis, Distribution Analysis, and Performance Simulation.

* System Generation

System Generation translates the system design information, held on the encyclopaedia, into a fully functional application system. Additional inputs from the encyclopaedia are technical information held on the encyclopaedia detailing the target environment for the system to be generated. This allows the design specification to be stored in a manner independent of the target hardware and software configuration. This approach also permits easy transfer of systems produced in this manner between different manufacturer's hardware.

* Project Management.

The central encyclopaedia is the single source of information about projects and their progress through the stages of the development cycle. Project and user managers and quality control or audit staff can therefore monitor progress against target criteria automatically. This sub-architecture can also support resource scheduling, task allocation, forecasting, productivity assessment through function point measurement [1] and ad-hoc reporting.

* Environment Control

A viable set of tools, in addition to being integrated about a 'core' encyclopaedia, should support the distributed working environment of systems planners, analysts and designers, such as that illustrated in figure 13.

If the workstation itself is portable, then it will allow the analyst to work alongside users, to create the models that will form the basis of all future design work. This portability, added to the use of graphic modelling techniques, will give analysts a medium for communication with users that has long been needed.

The requirement for distributed workstation support creates a set of problems that the architecture must cope with. For example, where separate project teams are to work on and update different, but overlapping, portions, of a single corporate entity relationship model, central co-ordination and access control is needed to prevent clashes.

The architecture allows the selection of subsets of higher level models for detailed study and provides independent version control of each subset to allow many project teams to work simultaneously.

The use of subset project models allows updates to the centrally held global model to be intercepted and rationalised by a project manager before those updates are authorised by the central coordinator. The architecture therefore incorporates communications control between workstation and host.

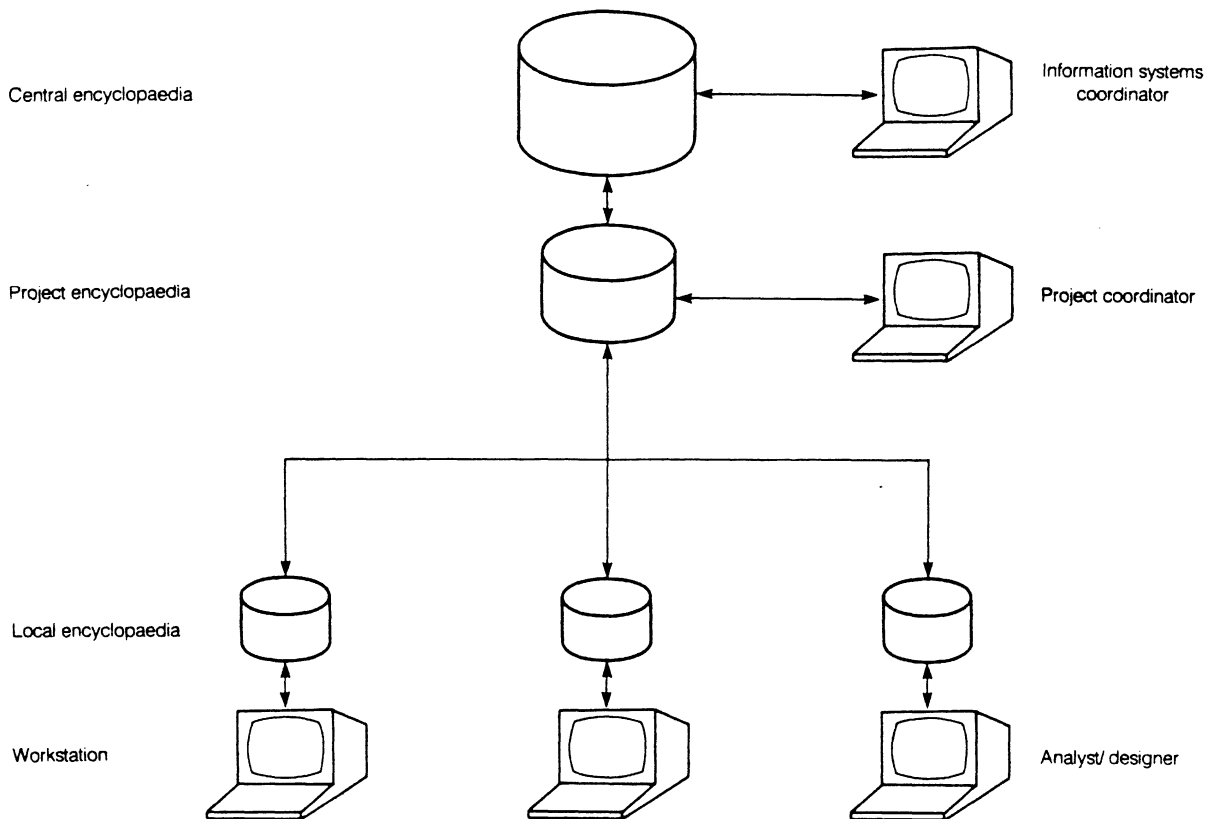


Figure 13: The Distributed Workstation Environment.

3.4.3 THE INFORMATION ENGINEER GRAPHICS AND MODELLING TOOLS

Combining existing Computer Aided Design technology with the encyclopaedia, allows the creation of a set of graphics based tools that allow not only interactive modelling, but also creation and maintenance of the encyclopaedia objects.

These tools require personal computer workstations using high resolution colour graphics and a pointing device, such as a mouse, as a suitable human diagramming interface.

The diagrams should not be stored in a graphical or pseudo-graphical form but in the form of encyclopaedia objects. The tools should be able to store and re-create diagrams using only these objects. This allows not only automatic diagramming, using optimal object placement and connection routing algorithms, but also allows overview (abstraction) and subset diagrams to be created.

Information Engineering makes use of eleven principal diagram types. These are used at different levels of detail during the analysis and design stages of the life cycle.

- decomposition diagrams
- entity relationship diagram
- dependency diagram
- data navigation diagrams
- state transition diagrams
- decision tree diagram
- data analysis diagram
- action diagrams
- data flow diagram
- data structure diagrams
- input/output layouts

As mentioned earlier, these diagrams use a consistent 'syntax', ie. symbols of a given shape have the same meaning across all diagrams. This simplifies the user interface and the interface with the encyclopaedia. Tools that incorporate these diagrams will therefore allow information systems personnel to share a common language, allowing their disciplines to begin to merge.

These diagramming techniques are combined to create the tools required at each stage of the methodology, so that any one tool may make use of one or more of the diagrams.

Figure 14 shows how the tools are integrated within the Encyclopaedia Maintenance architecture and also shows some of the interfaces to the System Generation architecture.

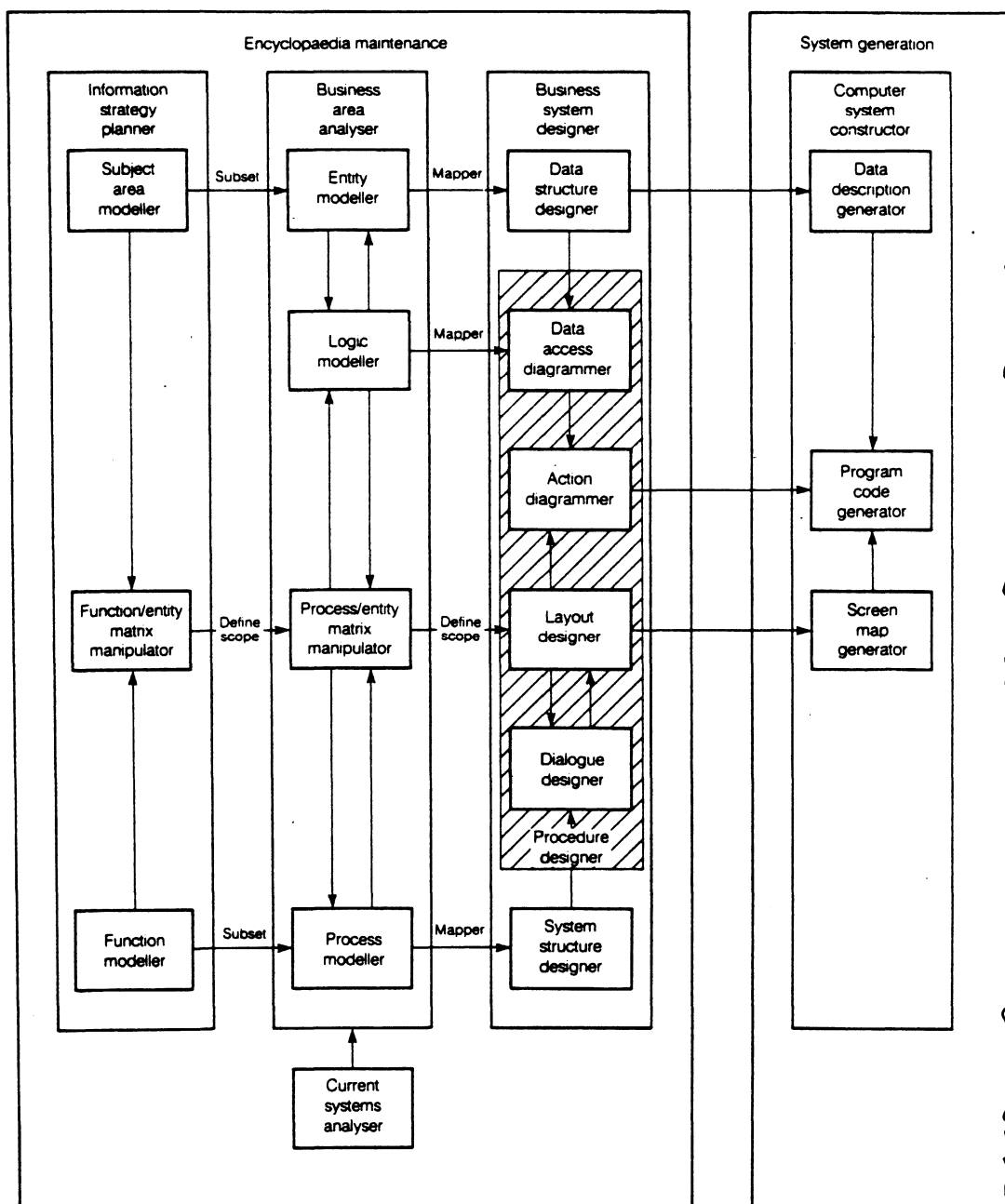


Figure 14: Integration of the Tools Within the Information Engineer

4. AUTOMATING THE IFIP PROBLEM

During the past four years the methodology has been thoroughly revised and defined in considerable detail. During the past two and a half years substantial effort has been extended on providing automated support. This effort is not complete so in this section the approach to automation is illustrated using trial versions of tools that are being integrated within a comprehensive workbench.

The software has been constructed by Texas Instruments and is used with their permission. It is demonstrated by TI in the James Martin Seminars but is not an announced product.

4.1 ENTITY ANALYSIS

The starting point for the IFIP problem is the entity relationship diagram. This remains essentially as for the original paper although some further generalisation has been undertaken. IFIP Unit therefore replaces the various committees and organisations and Role replaces Authorship and Refereeship.

Figure 15 shows how this appears on the entity relationship diagrammer. This, like other diagrams in this section is a print of the screen image from the TI Portable Professional Computer. In fact this diagram, like all of the others, is in full colour on the screen so some of its true value is lost here. It is important to note that this diagram (and the others) is a means of input. The nature of each object and the properties of objects that appear on the screen are understood by the system, captured and stored in its encyclopaedia. Once the information is captured diagrams are generated as needed from encyclopaedia contents. Diagrams are not stored just as graphics.

From this diagram it is possible to jump to other types of diagram which provide further detail for other aspects of the entity types. The first of these is the entity hierarchy diagrammer which represents subtype partitionings. Figure 16 shows the partitioning of Role into Authorship and Refereeship and the partitioning of Person into External Expert and IFIP Member. The hierarchy lines are coloured differently to relationship lines. At this level, relationships unique to particular subtypes can be shown.

The next level of detail is the data analysis diagram as shown in figure 17. This illustrates the attributes of an entity type and shows the identifying attribute(s) with its functional dependents. Here the complete screen layout is shown.

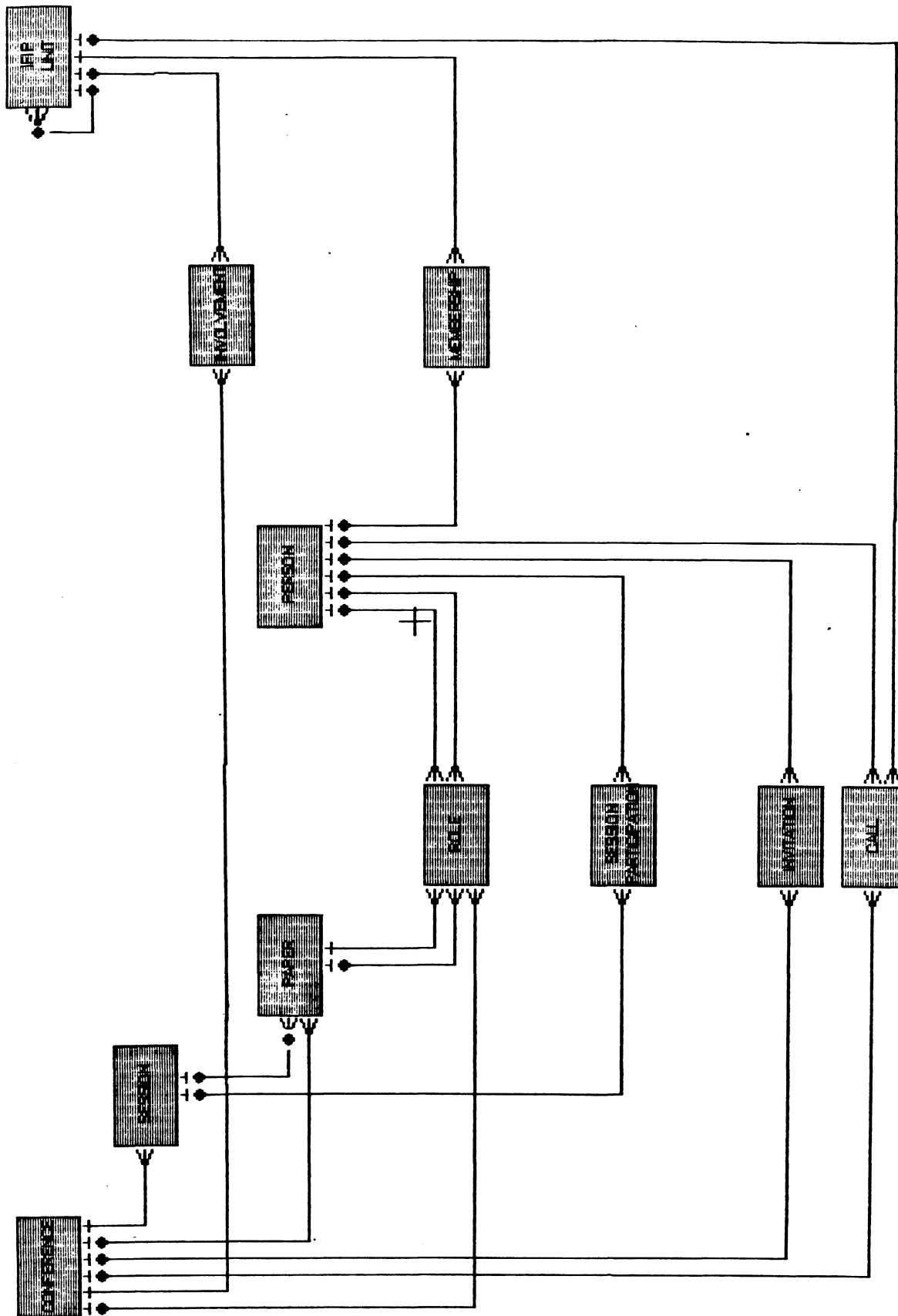


Figure 15. The Entity Relationship Diagram

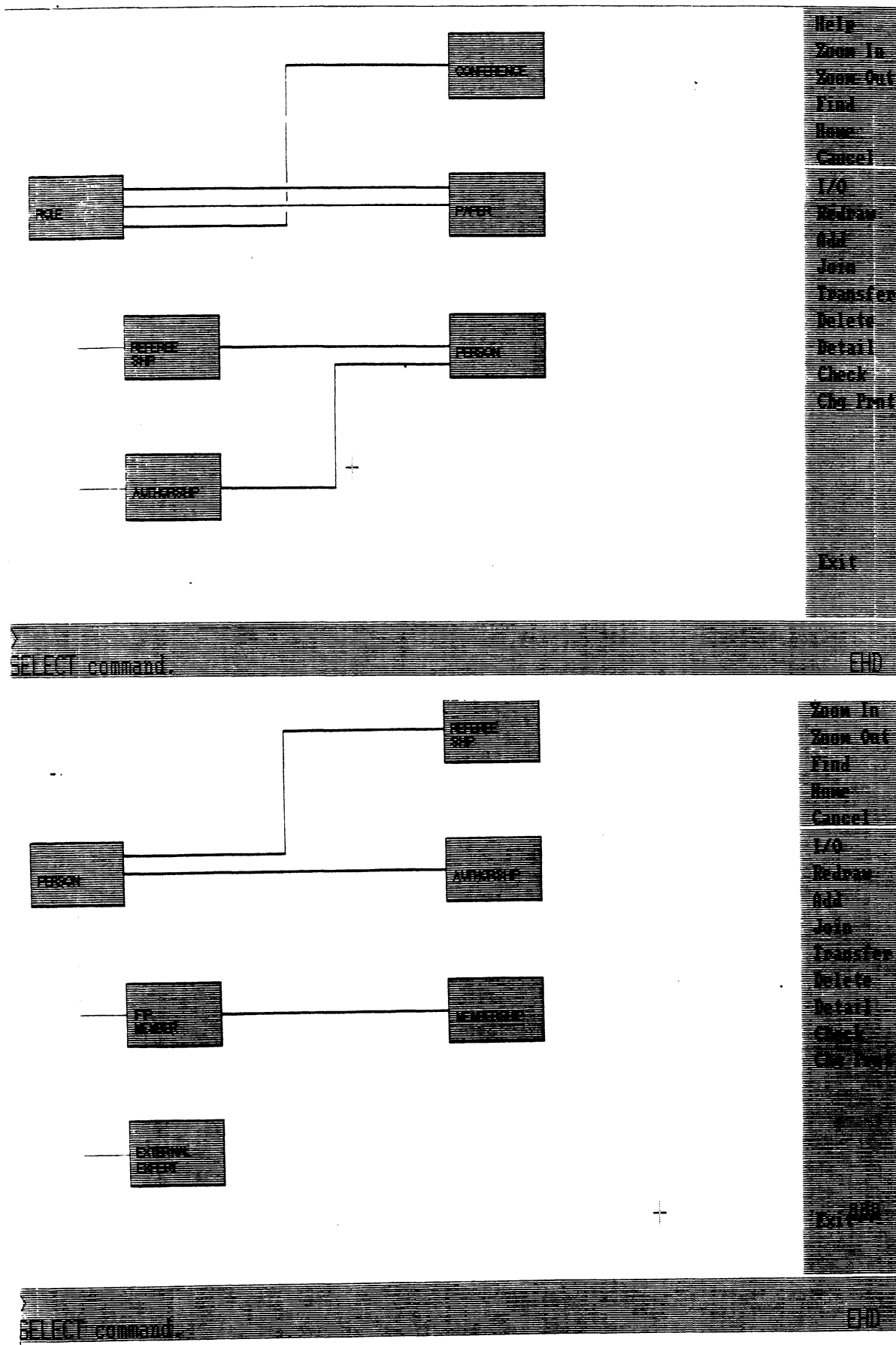


Figure 16: Entity Subtypes for Role and Person

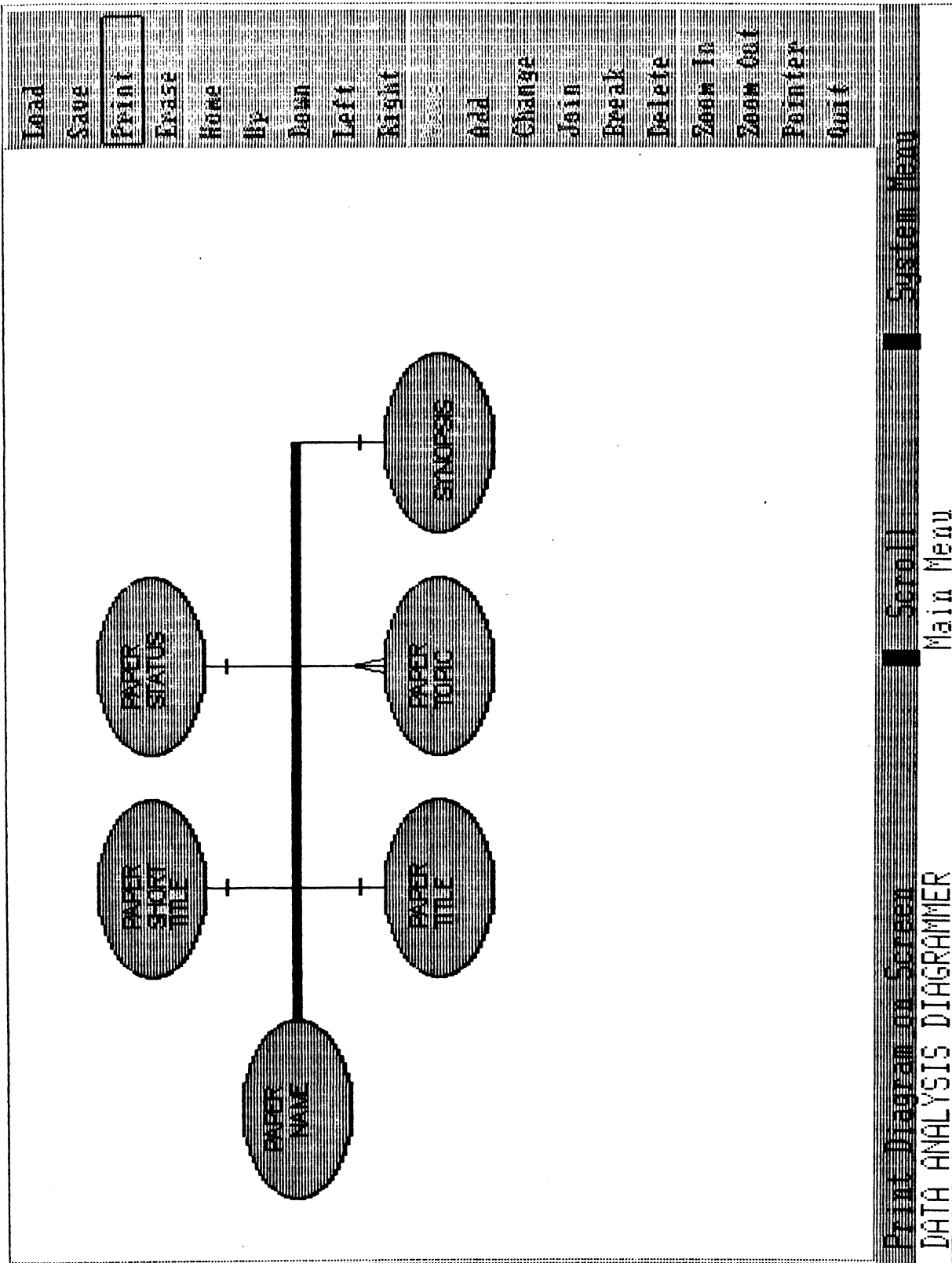


Figure 17

Data Analysis Diagram for Attributes of Paper

4.2 FUNCTION ANALYSIS

General functions are decomposed into specific business processes. This is illustrated in the form of a hierarchical decomposition diagram as shown in figure 18. Note that the crow's foot is used to indicate where multiple executions of a process may occur within a function. Other standard symbols could also be incorporated. Once the set of elementary business processes has been found by this means, a more rigorous representation, the process dependency diagram, is constructed. It shows the associations between them and shows their information requirements. This diagram is illustrated in Figure 19. It contains the processes, dependencies between them, the names of the information views on which the dependencies are based and features of the dependency such as cardinality of execution, optionality and associated conditions. Triggering events may also be shown, eg. WG Meeting triggers Conceive Conference.

4.3 INTERACTION ANALYSIS

Now that a clear understanding of data and activity is available, further detail can be added by analysing specific interactions between the two. This is done from opposing viewpoints, first by examining all processes which use details of entities of one type, then by examining all entity types required to satisfy the information needs of one process. The first is entity type life cycle analysis. This is illustrated in the form of a state transition diagram as in figure 20, where each state that entities of the Paper type can enter are shown as vertical bars while the transitions possible from each state and the conditions governing them are shown horizontally. The knowledge of what processes are valid in each state will be used in the final system to control the execution of transactions.

The second technique is process logic analysis. This is illustrated in the form of a navigation chart through a subset of the entity relationship diagram. Figure 21 illustrates the subset that might be needed by the Register Contribution process. This is selected then placed into background mode. The entity types required by the process, principal actions carried out by it, and the main sequence of actions are then highlighted on the diagram, resulting in a picture such as that in figure 22. This allows us to clarify our views on the nature of the process and to confirm that the data model is adequate. It can also be automatically transformed into a structured representation of the logic - an action diagram. The result is shown in figure 23. Further detail, eg. of conditions or of actions on attributes, can be edited into this diagram which could now, say in an information centre environment, be turned into code in a very high level language with little effort and no real design work.

With increasingly high level languages, this approach will become commonplace and good business models will supplant design specifications.

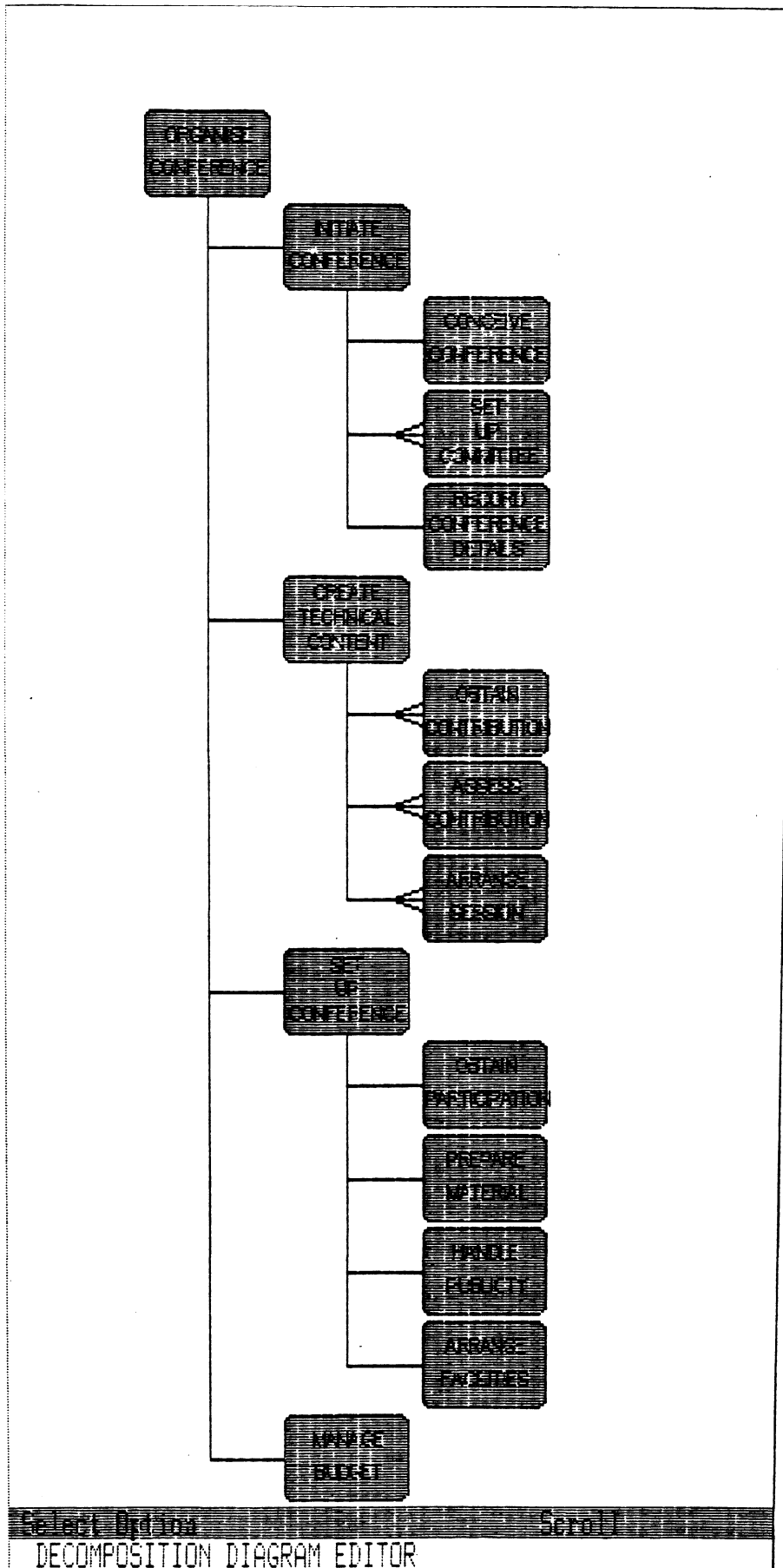
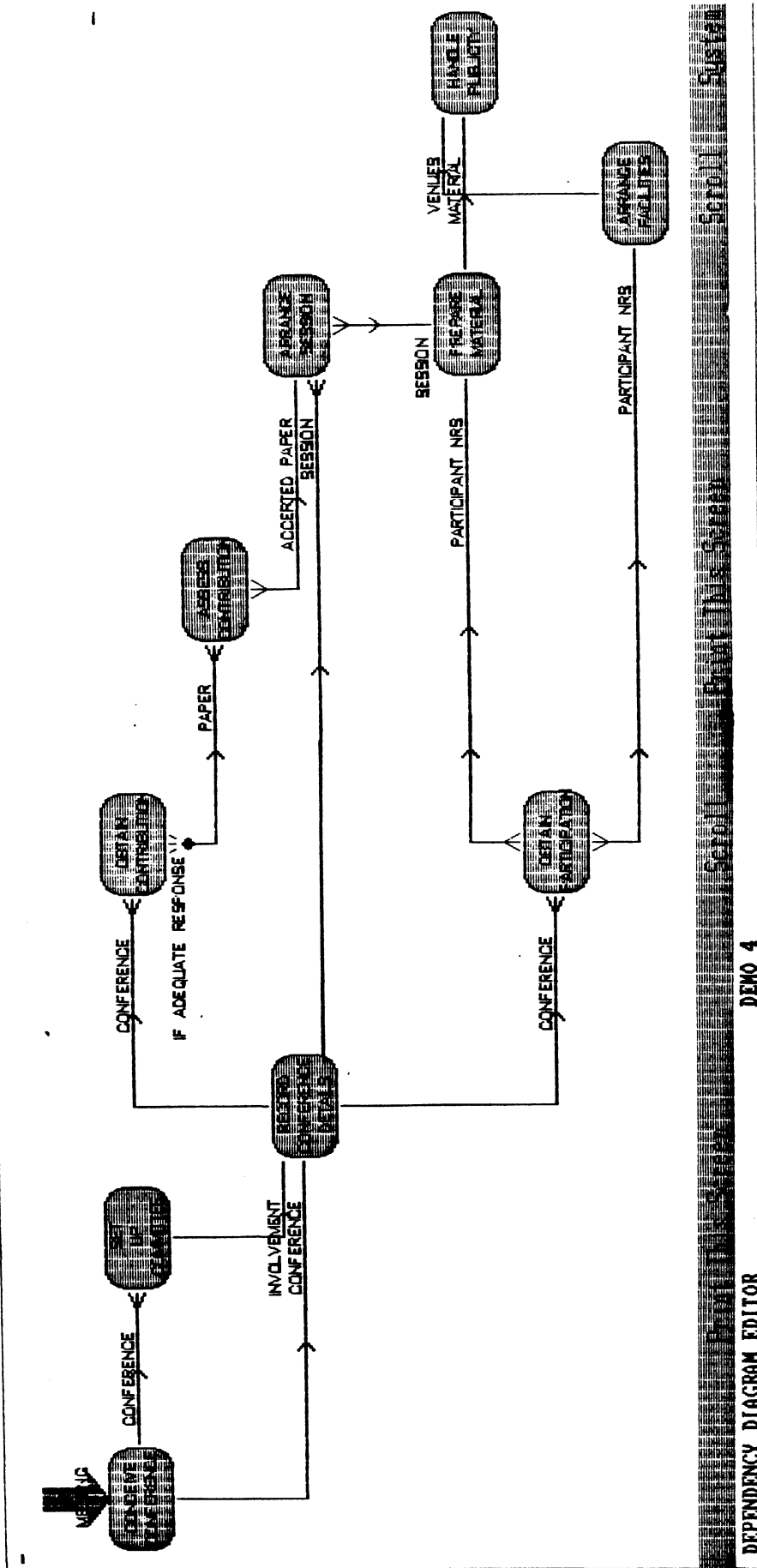


Figure 18: The Function Decomposition



DEMO_4

DEPENDENCY DIAGRAM EDITOR

Figure 19: The Process Dependency Diagram

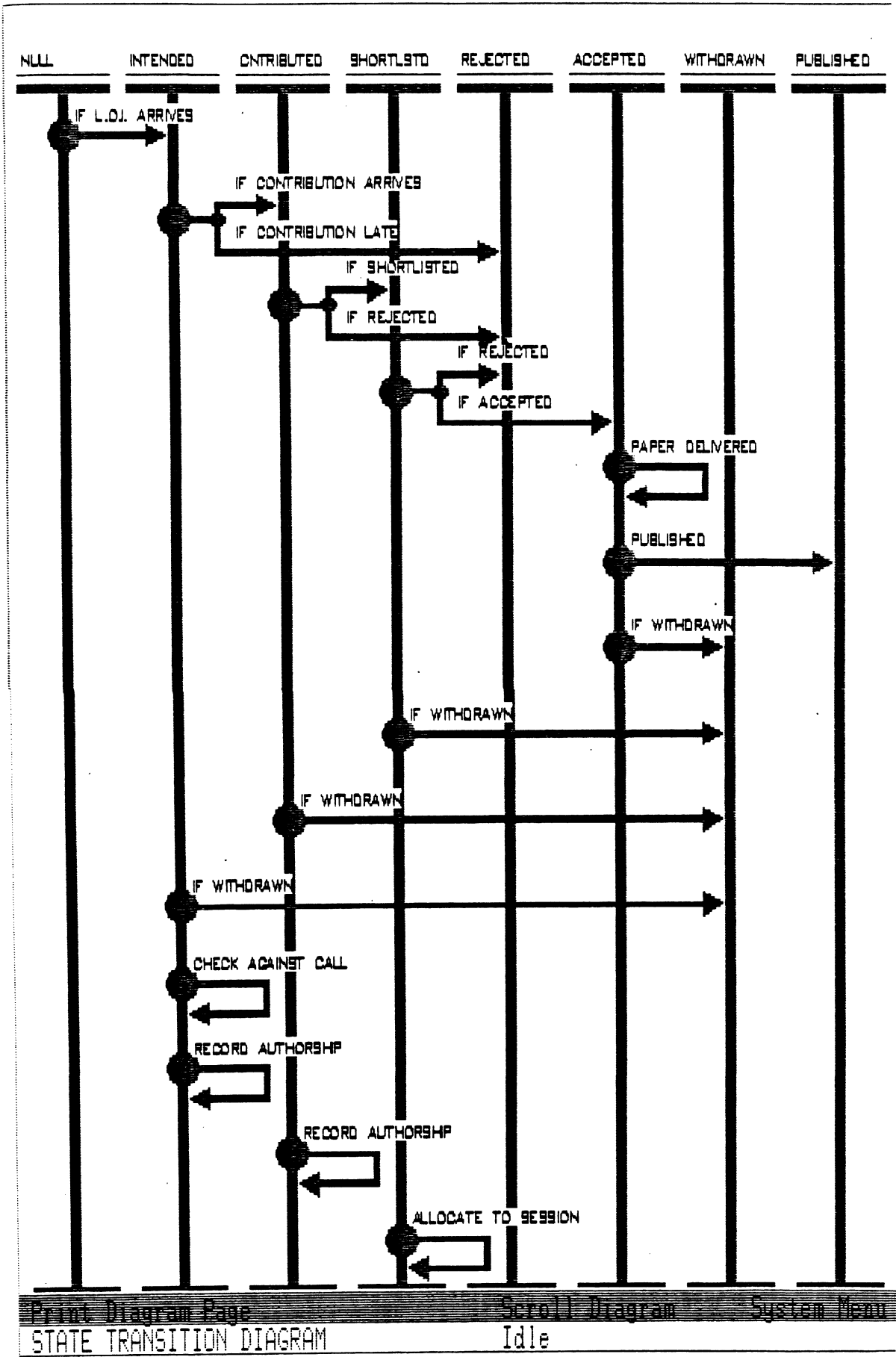


Figure 20: The Life Cycle for the Paper Entity Type

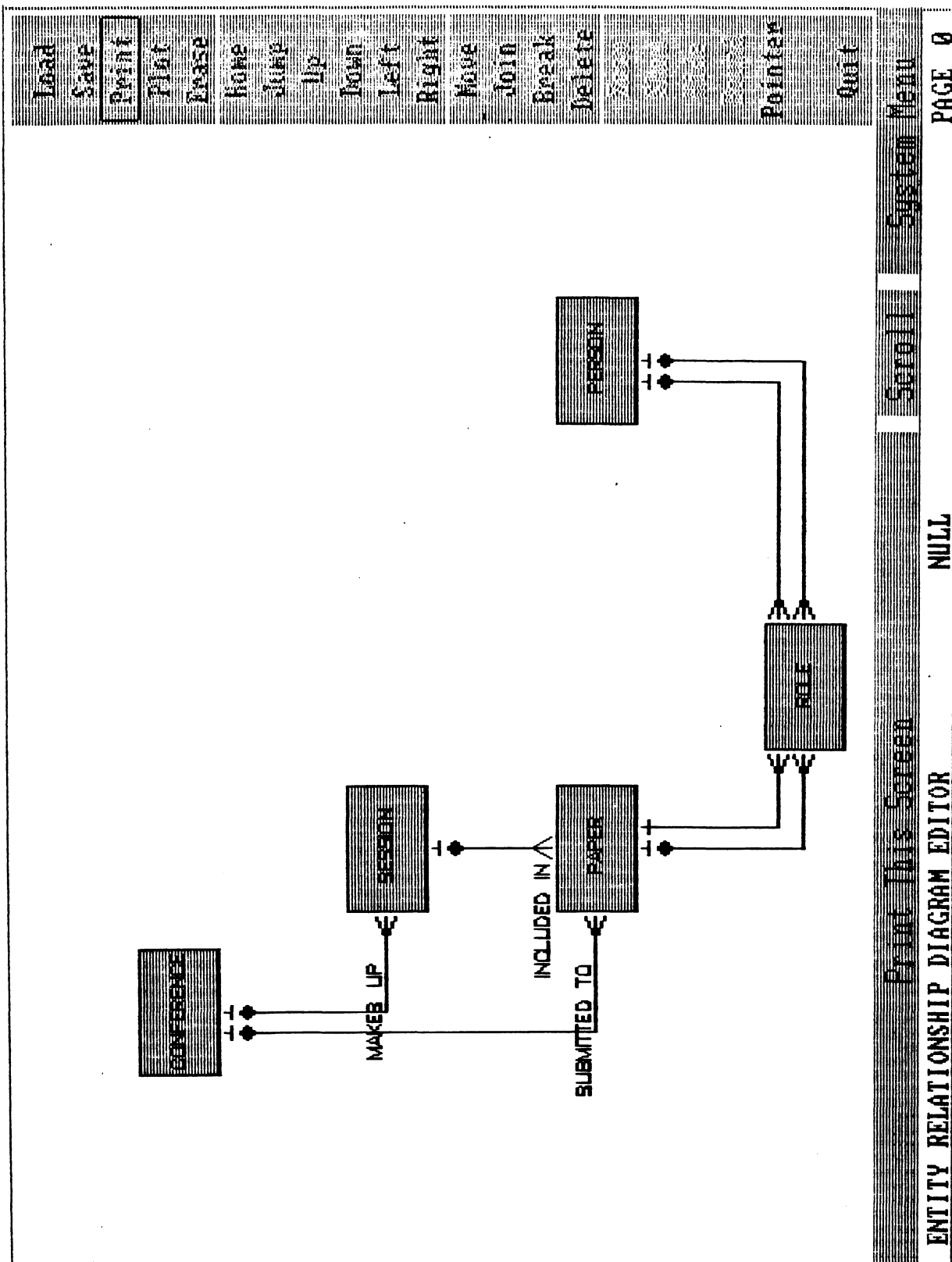


Figure 21: Entity Relationship Diagram Subset for Register Contribution

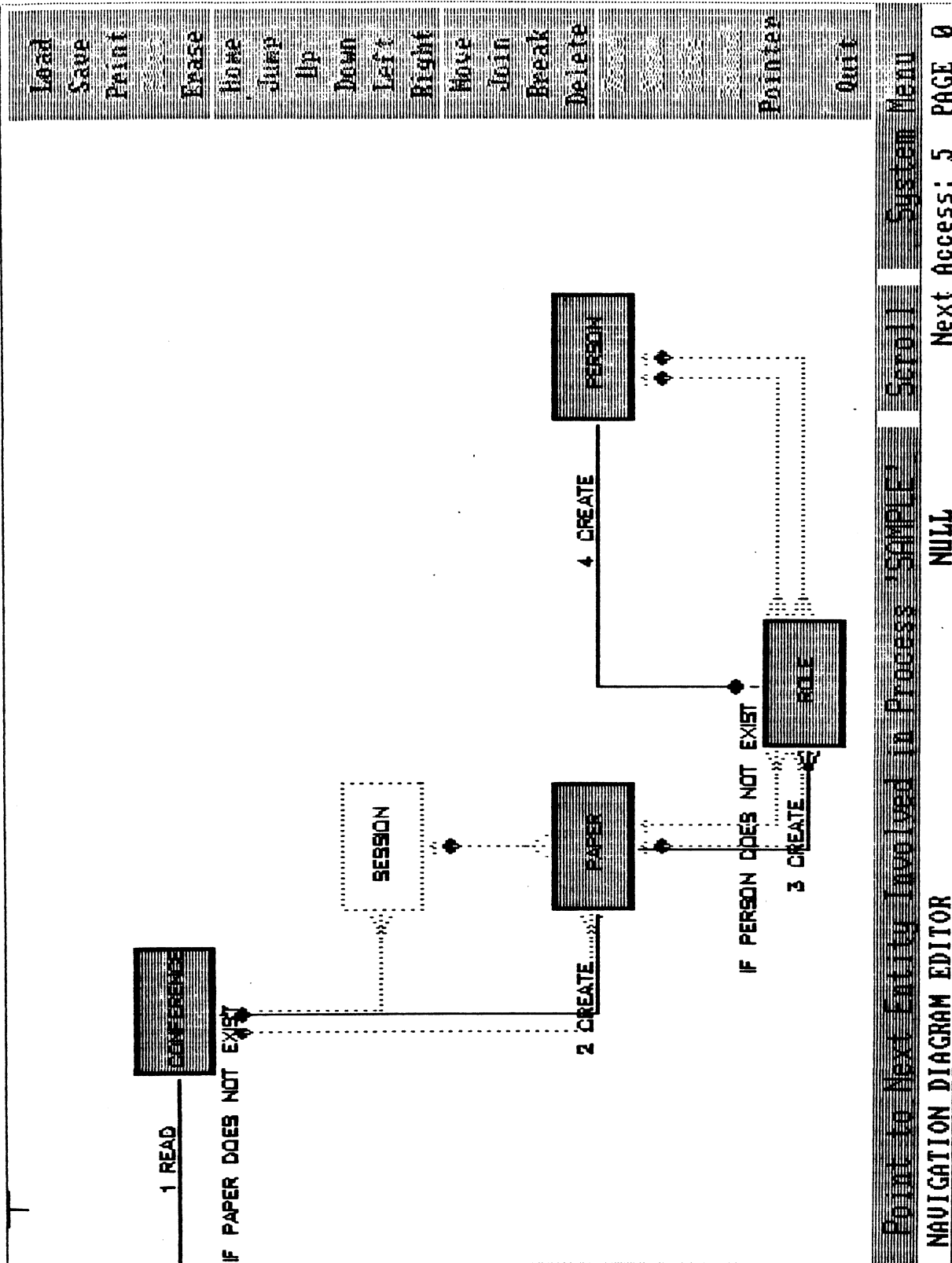


Figure 22: Process Logic Diagram for Register Contribution

```

PROC SAMPLE
READ CONFERENCE
IF PAPER DOES NOT EXIST
CREATE PAPER
DO WHILE <condition>
CREATE ROLE
IF PERSON DOES NOT EXIST
CREATE PERSON
ENDIF
ENDDO
ENDIF
ENDPROC

```

- Load
- Save
- Print
- Up
- Down
- Insert
- Delete
- Move
- Copy
- Repeat
- Submenu
- Esc
- F1-F6
- Do While
- Do Until
- Del End
- Exec
- Process
- Quit

Print Diagram System Menu

File Save Complete

Figure 23: Process Action Diagram for Register Contribution

4.4 DESIGN AREA DEFINITION

Once analysis is complete, the scope of each system must be defined. This can be given automated support through cluster analysis. An entity type/process matrix is constructed and the principal actions (Create, Modify, Use) recorded in the cells. A clustering algorithm is applied to group the entity types into potential data stores and the processes into potential systems. The matrix is then reorganised to show the combined effect. Figure 24 shows results obtained using Ward's Method [16]. This matrix is not at a sufficiently detailed level to give fully credible results but is quite sufficient to illustrate the principle and to highlight possible sub-systems and their interaction (interfaces or data flows).

The overall structure of the system can then be illustrated by constructing a data flow diagram as in figure 25. This provides a logical sequence for development which is modified in the light of business requirements and migration effort to give an implementation sequence. The implementation sequence defines the order of subsequent design and construction projects.

4.5 BUSINESS SYSTEM DESIGN

The first task of Business System Design is to create a preliminary database design for the original business area. The mapping rules for the chosen data management system are applied to the entity relationship model. This is trivial in the case of Codasyl or Relational systems. For DL/1 it is slightly more elaborate, involving the identification of proper root segments and their natural children, then resolving multiple parentages and involutions by creating link segments. Figure 26 shows the preliminary DL/1 design for the entity relationship model in figure 14.

The system structure is then designed by mapping from the processes and dependencies of analysis to procedures and data flows within the system. This creates a data flow diagram at a further level of detail down from that shown in figure 24.

For each procedure, a prototyping tool is used to format its screens and reports; a state transition diagram (similar to figure 19) illustrates the flows within its dialogue; and the prototype allows users to experience its behaviour. The procedure's use of data is illustrated in a data access diagram, which is similar to a process logic diagram but based on the preliminary data structure. Like the process logic, the data access diagram can be automatically translated into an action diagram for the procedure. In principle, this procedure action diagram can then be translated into code. This is illustrated in figures 26 and 27. The action diagram (in this case, the one shown in figure 22) is presented, as source language, to a translator which holds the rules for translation as Lisp statements. Cobol has been selected as the object language with the results seen in figure 27. This simply illustrates the principle. In practise a full scale system generator is employed to create fully linked systems for a defined target environment. Moreover, it may be necessary to go through a Technical Design stage to optimise the physical database and program structures for heavy-duty applications.

(Sub-) System	Data Store	Entity Type	Process	Calls			Refereeships			Papers			Conference			Organisational		
				Call	Refereeship	Paper	Person	Authorship	Conference	Session	Invitation	Specialist	Membership	Involvement	IFIP	Unit		
				2	7	4	5	6	1	3	8	9	10	12	11			
Call Handling	Publish General Call			1	C				U									U
	Issue Specific Call			2	C			U	U									
Referee Admin.	Arrange Referee			3		C		U	U	U								
	Organise Refereeing			15		C	U	U	U									
Contribution Handling	Register Letter of Intent			11	U		C	C	C	U								
	Register Contribution			12	U		C	C	C	U								
	Record Change			13			M	M	M	U								
Papers Management	Publish Proceedings			9			M	U	U	U								
	Select Paper			14		U	M	U	U		U							
Conference Definition	Record Conference Details			7				U		C	C			U	U	U		
	Determine Session Content			8			U				C							
Conference Organisation	Arrange Invitation			4				U		U		C						
	Handle Response			5				U		U		U						
	Prepare Handouts			10		U	U	U	U	U	U	U	U					
	Determine Audience			16		U		U					U	U			U	
	Provide Special Participant			6				U		U	U	C						

Figure 24: Clustered Matrix with Sub-Systems and Data Stores

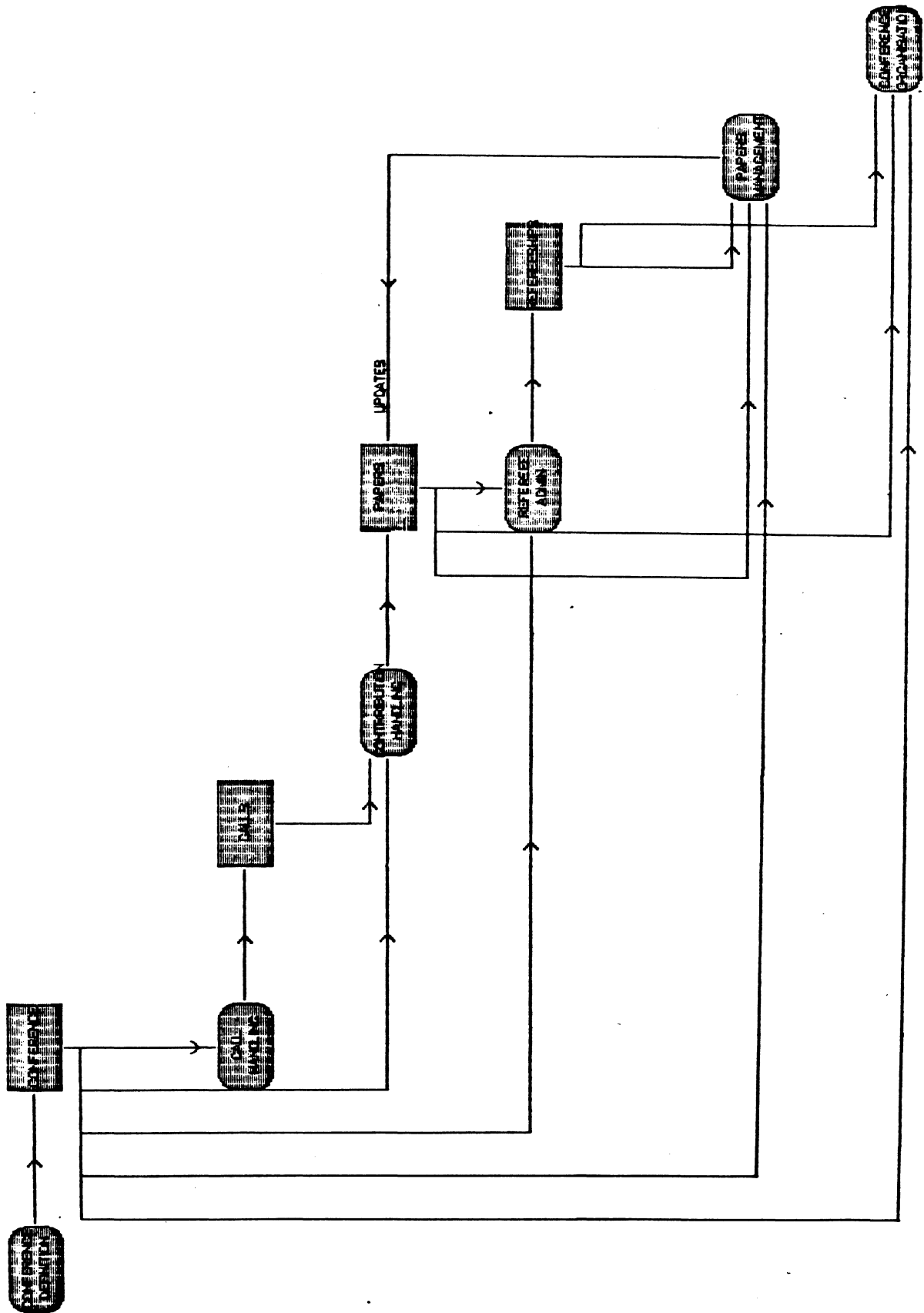


Figure 25: Data Flow Diagram for IFIP Sub-Systems

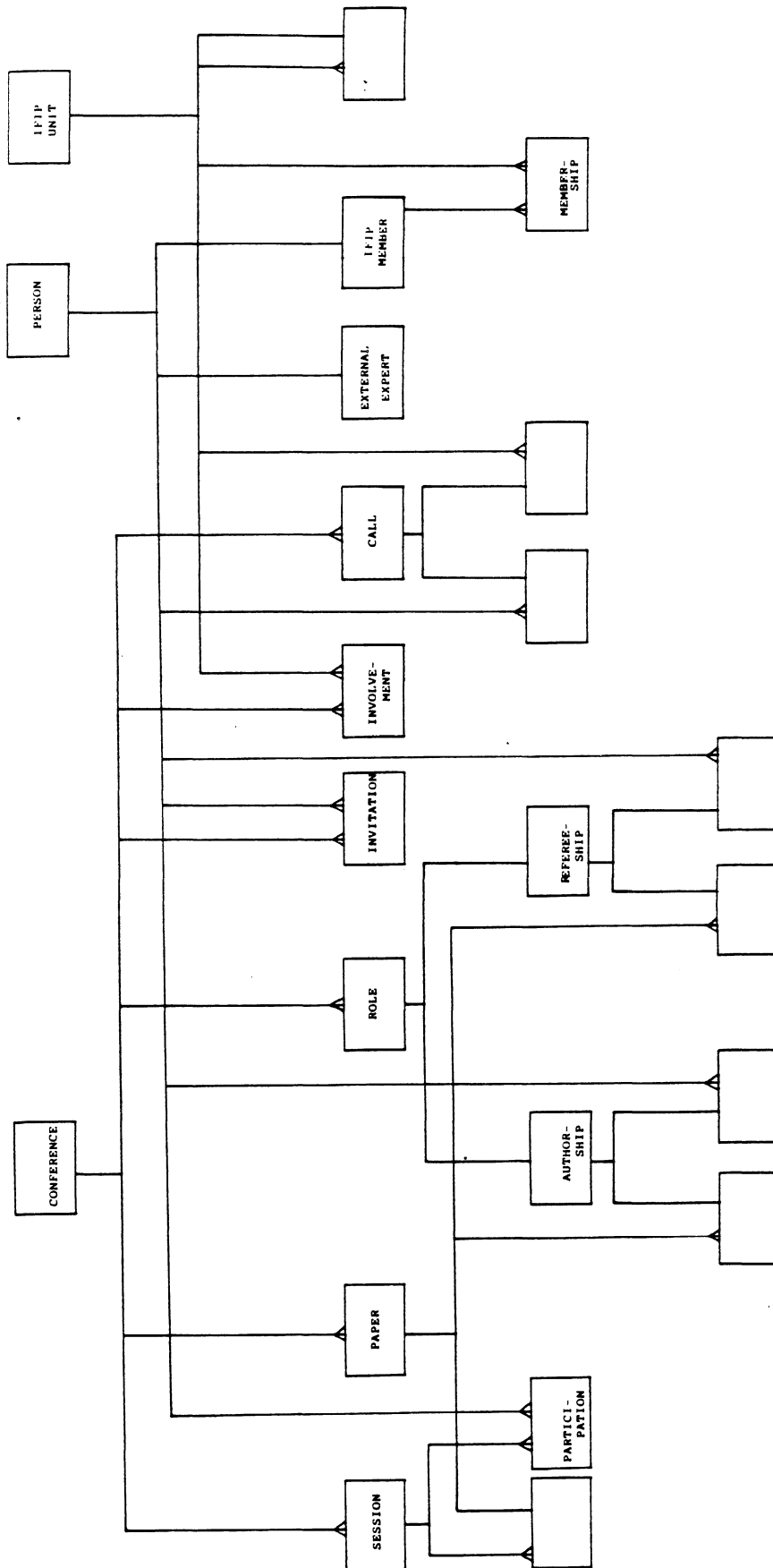


Figure 26: The Preliminary DL/1 Design

SOURCE LANGUAGE: Action Diagram Notation											
<pre> PROC SAMPLE READ CONFERENCE IF PAPER DOES NOT EXIST CREATE PAPER DO WHILE <condition> CREATE ROLE IF PERSON DOES NOT EXIST CREATE PERSON ENDIF ENDDO ENDIF ENDPROC </pre>											
OBJECT LANGUAGE: COBOL HIST 3.28 (1974)											
<pre> 000010 IDENTIFICATION DIVISION. 000020 PROGRAM-ID. SAMPLE. 000030 AUTHOR. IEF TRANSLATOR. 000040 INSTALLATION. 000050 DATE WRITTEN. 000060 DATA COMPILED. 000070 SECURITY. 000080 ENVIRONMENT DIVISION. </pre>											
Edit	Select	Load	Save	Go	Source	Object	Expand	Reduce	Lisp	Help	Exit

Figure 27: An Action Diagram Being Translated

```

SOURCE LANGUAGE: Action Diagram Notation
PRDC SAMPLE
OBJECT LANGUAGE: COBOL ANSI X3.23 (1974)
000010 IDENTIFICATION DIVISION.
000020 PROGRAM-ID. SAMPLE.
000030 AUTHOR. IEF TRANSLATOR.
000040 INSTALLATION.
000050 DATE WRITTEN.
000060 DATA COMPILED.
000070 SECURITY.
000080 ENVIRONMENT DIVISION.
000090 CONFIGURATION SECTION.
000100 SOURCE-COMPUTER. TIPC.
000110 OBJECT-COMPUTER.
000120 SPECIAL-NAMES.
000130 INPUT-OUTPUT SECTION.
000140 DATA DIVISION.
000150 FILE SECTION.
000160 WORKING-STORAGE SECTION.
000170 77 COPY COMMON-CMCB.
000180 LINKAGE-SECTION.
000190 77 CALLER-CMCB PIC X(128).
000200 PROCEDURE DIVISION.
000210 MAIN.
000220 CALL READ-CONFERENCE USING COMMON-CMCB
000230 A-0001.
000240     IF PAPER DOES NOT EXIST
000250     CALL CREATE-PAPER USING COMMON-CMCB
000260 A-0002.
000270     PERFORM A-0003 WHILE <condition>
000280 A-0003.
000290     CALL CREATE-ROLE USING COMMON-CMCB
000300 A-0004.
000310     IF PERSON DOES NOT EXIST
000320     CALL CREATE-PERSON USING COMMON-CMCB
000330 A-0004-EXIT.
000340 A-0003-EXIT.
000350 A-0002-EXIT.
000360 MAIN-EXIT.
000370     STOP RUN.

```

Edit	Select	Load	Save	Go	Source	Object	Expand	Reduce	Lisp	Help	Exit
------	--------	------	------	----	--------	--------	--------	--------	------	------	------

Figure 28: Cobol Generated from an Action Diagram

MAINTAINING THE MODELS

the automated support environment the encyclopaedia maintains information and is structured so that it has knowledge of significance of the constructs in each diagram type. It views its content as a single model with each diagram simply representing a subset of that model. It therefore maintains integrity over the whole model, eg. an attempt to edit an action on an action diagram for an entity type that was not in the original process logic will be treated as invalid until the process logic is updated.

only major mapping which occurs is between analysis objects and design objects. The initial assumption is that this will be one-to-one. Inevitably this is an oversimplification. Objects are allowed in the design which have no counterpart in the analysis, eg. access control permissions. The system is able to ignore such discrepancies automatically but human intervention provides the control.

Within the analysis objects, or the design objects, different levels of abstraction exist at different times to allow for representation at different levels. Control over these is provided by following the principles set out for data object grouping by Feldman and Miller [17] and for levelling of data flows and dependencies described by DeMarco [18] and by Hamilton and Zeldin [19,20].

Consistency is therefore preserved both internally and in various levels of external representation. Resulting systems are therefore of high quality and are implicitly documented to a very high standard.

CONCLUSIONS

Information Engineering is a major step forward from the D2S2 approach described in CRIS 1.

Information Engineering is designed to be automated and to take advantage of personal computer capabilities and advances in understanding of system generation techniques.

In scope, the methodology now fully covers all aspects of systems development. Its techniques have been refined and tightly integrated so that automation, based on an active encyclopaedia of developmental and systems knowledge, has become practical. This refining has been matched by a rationalisation and simplification of the diagramming conventions. The overall result is a marked improvement in rigour.

REFERENCES

- [1] Albrecht, A.J., AD/M Productivity Measurement and Estimate Validation, internal publication by IBM Corporate Information Systems and Administration, Purchase, N.Y., USA, (1984)
- [2] Davenport, R.A., Data Analysis for Database Design, Australian Computer Journal 10 (1978) 122-137.
- [3] Davenport, R.A., The Application of Data Analysis: Experience With The Entity Relationship Approach, in: Chen, P.P., (ed), Entity Relationship Approach to Systems Analysis and Design (North-Holland, Amsterdam, 1980).
- [4] Gibson, W.E. and Macdonald, I.G., The Corporate Database - Automating the Development Process, in: Iggulden, D., (ed), The Corporate Database (Pergamon Infotech, Maidenhead, England, 1985).
- [5] Macdonald, I.G., Information Engineering - A Methodology To Match Fourth Generation Tools, in: Tozer, E.E., (ed), Application Development Tools (Pergamon Infotech, Maidenhead, England, 1984).
- [6] Macdonald, I.G. and Palmer, I.R., System Development in a Shared Data Environment - the D2S2 Methodology, in: Olle, T.W., Sol, H.G. and Verriijn-Stuart, A.A., (eds), Information Systems Design Methodologies: A Comparative Review (North-Holland, Amsterdam, 1982).
- [7] Martin, J., Strategic Data-Planning Methodologies (Prentice-Hall Inc., Englewood Cliffs, N.J., USA, 1982).
- [8] Martin, J., Application Development Without Programmers (Prentice-Hall Inc., Englewood Cliffs, N.J., USA, 1982).
- [9] Martin, J., Managing the Data Base Environment (Prentice-Hall Inc., Englewood Cliffs, N.J., USA, 1983)
- [10] Martin, J., An Information Systems Manifesto (Prentice-Hall Inc., Englewood Cliffs, N.J., USA, 1984)
- [11] Martin, J., Recommended Diagramming Standards for Analysts and Programmers: A Basis for Automation (Prentice-Hall Inc., Englewood Cliffs, N.J., USA, 1986, in press).
- [12] Mitchell, W., Information Engineering, in: Data Analysis in Practice, proceedings of Database '85 (British Computer Society's Database Specialist Group, London, England, 1985).
- [13] Palmer, I.R., Practicalities in Applying a Formal Methodology to Data Analysis, in: Maddison, R.N. (ed), Data Analysis for Information System Design (British Computer Society, London, 1978).

- [14] Rock-Evans, R., Data Analysis (IPC Business Press, Surrey, England, 1981).
- [15] Shave, M.J.R., Entities, Functions and Binary Relations: Steps to a Conceptual Schema, Computer Journal 24 (1981) 42-47.
- [16] Murtagh, F., A Survey of Recent Advances in Hierarchical Clustering Algorithms, Computer Journal 26 (1983) 354-359
- [17] Feldman, P. and Miller, D., Entity Model Clustering: Structuring a Data Model by Abstraction, Computer Journal (in press), (1986).
- [18] DeMarco, T., Structured Analysis and System Specifications, (Yourdon Inc., New York, USA, 1978).
- [19] Hamilton, M. and Zeldin, S., Higher Order Software - A Methodology for Defining Software, IEEE Transactions on Software Engineering 2 (1976).
- [20] Hamilton, M. and Zeldin, S., The Relationship Between Design and Verification, Journal of Systems and Software 1 (1979) 29-56.

Information Systems and Organizational Change

Peter G. W. Keen
Sloan School of Management, MIT

This paper discusses long-term change in organizations in relation to information systems. It reviews causes of social inertia, resistance and counterimplementation. It stresses the pluralistic nature of organizations. Tactics for managing change rely on incremental, facilitative approaches. These limit strategic change which requires coalition-building and careful attention to political mechanisms.

Key Words and Phrases: implementation, change, politics

CR Categories: 2.11, 3.30

1. Introduction

This paper discusses long-term change in organizations in relation to information systems. The aim is to explain why innovation is so difficult and to point towards effective strategies for managing the process of change. Many commentators have drawn attention to the problems of implementation that result in systems being technical successes but organizational failures. (Urban [69], Grayson [23], Keen [32], Drake [19]) Their analyses stress the complexity of organizational systems and the social inertia that damps out the intended effects of technical innovations.

The growing body of research on implementation deals mainly with tactical issues: How to create a climate for changing, building and institutionalizing a specific system.¹ This paper focuses on strategic questions:

- (1) What are the causes of social inertia?

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Author's present address: Peter G. W. Keen, Center for Information Systems Research, Sloan School of Management, Massachusetts Institute of Technology, Room E53-420, 60 Memorial Drive, Cambridge, MA 02139.

© 1981 ACM 0001-0782/81/0100-0024 \$00.75.

- (2) What are the main organizational constraints on change?
- (3) What are the mechanisms for effecting change?

Effective implementation relies on incremental change, small-scale projects, and face-to-face facilitation. (Ginzberg [22], Vertinsky et al. [71], Keen and Scott Morton [36]) A strategy for long-term change and large-scale innovation requires a broader strategy; the conceptual and empirical work on implementation, both within MIS and OR/MS and in political science, provides few guidelines and some very pessimistic conclusions. The main argument of this paper is that information systems development is an intensely political as well as technical process and that organizational mechanisms are needed that provide MIS managers with authority and resources for negotiation. The traditional view of MIS as a staff function ignores the pluralism of organizational decision making and the link between information and power. Information systems increasingly alter relationships, patterns of communication and perceived influence, authority, and control. A strategy for implementation must therefore recognize and deal with the politics of data and the likelihood, even legitimacy, of counterimplementation.

2. The Causes of Social Inertia

"Social inertia" is a complicated way of saying that no matter how hard you try, nothing seems to happen. The main causes of inertia in relation to information systems seem to be:

- (1) Information is only a small component of organizational decision processes;
- (2) Human information-processing is experiential and relies on simplification;
- (3) Organizations are complex and change is incremental and evolutionary; large steps are avoided, even resisted;
- (4) Data are not merely an intellectual commodity but a political resource, whose redistribution through new information systems affects the interests of particular groups.

Computer specialists generally take for granted that information systems play a central role in decision making. Mintzberg's [51] and Stewart's [65] descriptive studies of managers' activities suggest this is often not the case. (see also Kling [39]) In general, decision processes are remarkably simple (Miller [50]); what has worked in the past is most likely to be repeated. Under pressure, decision makers *discard* information, avoid bringing in expertise and exploring new alternatives (Wilensky [75]); they simplify a problem to the point where it becomes manageable. Almost every descriptive study of a complex decision process suggests that formal analysis of quanti-

¹ See [34] for a critical evaluation of implementation research.

fied information is, at best, a minor aspect of the situation. (Pettigrew [57], Bower [7]) Negotiations, (Strauss [67]) habit, rules of thumb, and "muddling through" (Lindblom [46]) have far more force. This may seem an extreme assertion but there is little if any empirical evidence to challenge it. The point is not that managers are stupid or information systems irrelevant but that decision making is multifaceted, emotive, conservative, and only partially cognitive. Formalized information technologies are not as self-evidently beneficial as technicians presume. Many descriptive models of decision making (Lindblom [46], Cohen and March [12], Hirschman [28]) imply that "better" information will have virtually no impact.

Simon's concept of bounded rationality stresses the simplicity and limitations of individual information processing.² There has long been a conflict between the normative perspective of OR/MS and MIS, which defines tools based on a rationalistic model of decision making, and the descriptive, largely relativistic position of many behavioral scientists who argue that that conception is unrealistic.³ Mitroff's study of the Apollo moon scientists is perhaps the best supported presentation of this position. [54] Regardless of one's viewpoint on how individuals *should* make decisions, it seems clear that the processes they *actually* rely on do not remotely approximate the rational ideal. This gap between the descriptive and prescriptive is a main cause of inertia:

- (1) There is little evidence to support the concept of consistent preference functions (Braybrooke and Lindblom [9], Kahneman and Tversky [31], Kunruether and Slovic [42]);
- (2) Managers and students (the traditional subjects of experiments) have difficulty with simple trade-off choices (Zionts and Wallenius [77]);
- (3) Perceptions are selective (Dearborn and Simon [14]);
- (4) There are clear biases and personality differences in problem-solving "styles" (Huysmans [30], McKenney and Keen [49], Doktor [16]) that may even lead individuals to reject accurate and useful information (Churchman [11], Doktor and Hamilton [17]);
- (5) Even intelligent and experienced decision makers make many errors of logic and inference (Tversky and Kahneman [68], Ross [61]);
- (6) Managers prefer concrete and verbal data to formal analysis. (Mintzberg [51], Stewart [65]);

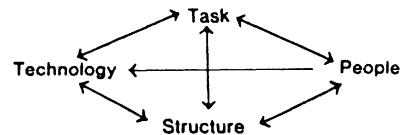
All in all, human information-processing tends to be simple, experiential, nonanalytic, and on the whole,

² See [64]. See also [13], and with a different flavor and very different conclusions, Lindblom [47] who argues that:

"The human condition is small brain, big problems. People then need help—devices, processes and institutions—to simplify problem-solving." (p. 66).

³ See [33] for a historical summary of the (largely axiomatic) concept of optimality.

Fig. 1. The Leavitt "Diamond": Components of the organization.



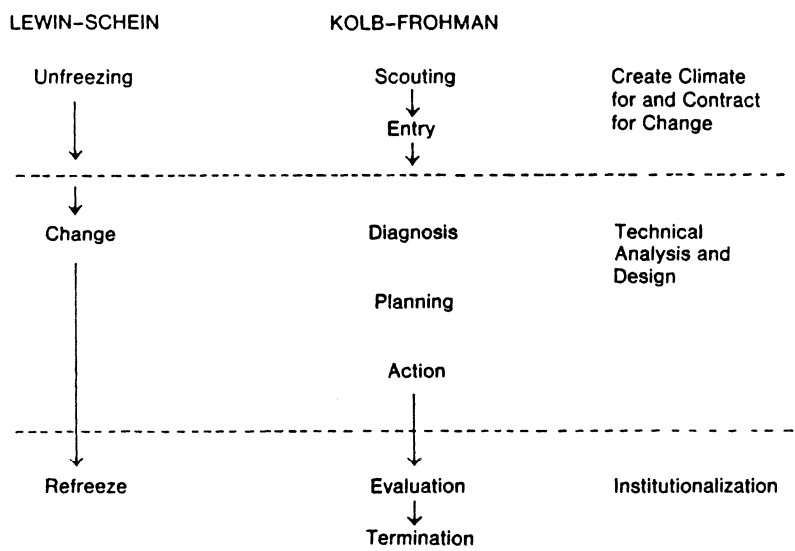
fairly effective. (Bowman [8], Lindblom [46]) Formalized information systems are thus often seen as threatening and unneeded. They are an intrusion into the world of the users who see these unfamiliar and nonrelevant techniques as a criticism of themselves.

Leavitt's classification of organizations as a diamond, (Figure 1) in which Task, Technology, People, and Structure are interrelated and mutually adjusting, indicates the complex nature of social systems. [44] When Technology is changed, the other components often adjust to damp out the impact of the innovation. Many writers on implementation stress the homeostatic behavior of organizations (Roberts [60], Ginzberg [22], Zand and Sorenson [76]) and the need to "unfreeze the status quo." (This term is taken from the Lewin-Schein framework of social change, discussed below.)

Information systems are often intended as coupling devices that coordinate planning and improve management control. (Galbraith [21]) Cohen and March's view of many organizational decision processes as a garbage can [12] and Weick's powerful conception of "loose coupling" [72] imply, however, that signals sent from the top often get diffused, defused, and even lost, as they move down and across units whose linkages are tenuous. The more complex the organization, the less likely the impact of technical change; homeostatic, self-equilibrating forces in loosely coupled systems are a major explanation for the frequency of failure of large-scale planning projects. (Hoos [29], Keen [32], Hall [25])

The characteristics of individuals and organizations listed above suggest that dramatic change rarely occurs in complex social systems. Lindblom's well-known concept of muddling through reinforces that view. [46] He points out the value of incremental, remedial decision making and rejects the "synoptic ideal." Wildavsky [74] similarly disdains formalized planning and recommends an avowedly political process based on partiality and incremental analysis. He contrasts political and economic rationality. The latter looks for optimal solutions through systematic methodologies. Compromise is pathological since by definition it represents a retreat from rationality (one might expect that few people would espouse this position in so pristine a form—until one listens to a faculty full of microeconomists). Political (or social) rationality looks only for feasible solutions and recognizes that utopian change cannot be assimilated by complex systems composed of individuals with bounded rationality. Only small increments are possible and compromise, far from being bad, is an essential aspect of the implementation process.

Fig. 2. Tactical Model for Describing and/or Managing Change.



The final cause of inertia is less passive than the others. Data are a central political resource. Many agents and units in organizations get their influence and autonomy from their control over information. They will not readily give that up. In many instances new information systems represent a direct threat and they respond accordingly. We now have adequate theories of implementation. We have less understanding of counterimplementation, the life force of more than a few public sector organizations and a hidden feature of many private ones. This issue is discussed in more detail in Section 6.

All these forces towards inertia are constraints on innovation. They are not necessarily binding ones. Implementation is possible but requires patience and a strategy that recognizes that the change process must be explicitly managed. Only small successes will be achieved in most situations. These may, however, be strung together into major long-term innovations. "Creeping socialism" is an instance of limited tactical decisions adding up to strategic redirection; no one step appears radical.

3. Overcoming Social Inertia: A Tactical Approach

There are several well-defined tactical models for dealing with inertia. They are tactical in the sense that they apply largely to specific projects. They recommend simple, phased programs with clear objectives (Pressman and Wildavsky [59]) and facilitation by a change agent or a "fixer" (Bardach [5]), an actor with the organizational resources to negotiate among interested parties and make side payments. The Lewin-Schein framework and an extension of it, Kolb and Frohman's model of the consulting process [41], have been used extensively

by researchers on OR/MS and MIS implementation,⁴ both in descriptive studies (Ginzberg [22], Zand and Sorenson [76]) and prescriptive analysis. (Lucas and Plimpton [48], Keen [32], Urban [69]) This conception of the change process (see Figure 2) emphasizes:

- (1) The immense amount of work needed prior to design; change must be self-motivated and based on a "felt need" with a contract between user and implementer built on mutual credibility and commitment;
- (2) The difficulty of institutionalizing a system and embedding it in its organizational context so that it will stay alive when the designer/consultant leaves the scene;
- (3) The problem of operationalizing goals and identifying criteria for success.

This tactical approach is "Up-and-In" rather than "Down-and-Out". (Leavitt and Webb [45]) DO is based on direction from the top, lengthy design stages, and a formal system for planning and project management. UI relies on small groups, with face-to-face involvement and participative management. The design evolves out of the Entry process. (Kolb and Frohman [41])

Leavitt and Webb point out that UI works well for small projects. However, large-scale change requires an engineering approach to design that quickly encounters social inertia. The dilemma is that UI limits itself to feasible, incremental change while DO, the broader strategic process, is rarely successful. The tactical model needs extension; facilitation is not enough and social inertia is dangerously close to social entropy (Bardach [5]).

⁴ Ginzberg [22] provides a useful summary of this perspective. See also [34].

No formal effective strategic model exists. If it did, one might expect to find it in political science, which frequently reconstructs the processes underlying efforts to deliver major social, technical, or political programs. (Saplosky [63], Pressman and Wildavsky [59], Hargrove [27], Derthick [15]) Political science deserves the label of the "dismal" science far more than economics, which after all believes in the eventual triumph of rationality; most studies in this field deal with failures. (Saplosky's analysis of the Polaris project is a rare example of a success.) They identify as forces impeding change not only social inertia but also pluralism and counterimplementation—overt moves, often made by skilled actors, to prevent a disruption of the status quo. Counterimplementation is most likely to occur when outsiders bring in threatening new technologies. (Munson and Hancock [55], Chesler and Flanders [10]). Information systems are exactly that in many cases (Argyris [3], [4], Hall [25], [26].)

4. Pluralism: The Need to Mobilize

Political science views organizations mainly as groups of actors, often with conflicting priorities, objectives, and values. (Allison [1]) The management literature generally assumes far more commonality of purpose. The Down-and-Out approach relies on this. Up-and-In evades the problem by limiting the scope of the project and hence the number of actors involved; it fails completely if consensus is not impossible. The more the organization is viewed as a set of loosely coupled units (Weick [72]) where joint action rests on negotiations (Strauss [67]), the more any strategy for implementation must emphasize the need to mobilize coalitions, to provide the necessary support for an innovative proposal. Obviously, that process is based on political rather than economic rationality. The corollary of this argument is that lack of attention to the constraints on change imposed by pluralism in organizations will result in failure.

Many writers who attack the rationalist tradition on which OR/MS and MIS are based stress the legitimacy of pluralism and hence of incremental decision making. Lindblom sees the use of social interactions instead of analysis and planning as analogous to reliance on a market system to simplify the process of resource allocation [47]. Strauss argues that "social order" and decision making in any organization are predominantly based on negotiations:

... when individuals or groups or organizations work together to 'get things done' then agreement is required about such matters as what, how, when, where, and how much. Continued agreement itself may be something to be worked at ... negotiations pertain to the ordering and articulation of an enormous variety of activities.⁶

In many instances, pluralistic perspectives view formal information systems as either ethically dangerous in

⁶ [67], p. ix.

that they impose a false rationality (Hoos [29]), naive (Wildavsky [74]), or simply irrelevant. (Lindblom [46]) They also deny their value as coupling devices that help coordinate planning and communication; pluralists see *merit* in disorder and redundancy. (Klein and Meckling [37]) Weiner and Wildavsky, commenting on federalism, summarize this argument: What is needed is "... planning with a different aim: to foster choice through careful structuring of social interaction."⁶

These viewpoints are obviously not shared by most proponents of analytic methodologies. Since they are mainly based on studies of public policy issues, one may argue that business organizations are more tightly coupled and less dominated by pluralism and incrementalism. This may be true in particular instances; there are many companies whose planning systems are effective in establishing and communicating goals, involving managers in the decision process, and creating a climate for innovation. (Vancil and Lorange [70]) Even so, most case studies of complex decisions suggest that companies are far more pluralistic than we conveniently assume. Pettigrew's analysis of a decision to purchase a computer, for example, reveals innumerable territorial disputes, maneuvering for position, conflict over goals, and irreconcilable differences in perspective among organizational units [57]. Believers in pluralism do not find that surprising but most computer specialists do.

The point is not to justify pluralism. It seems clear, however, that it is a main cause of inertia. "Getting things done", whether Down-and-Out or Up-and-In, requires the careful building of coalitions based on complex negotiations. The larger the scope of a project and the more strategic its goals, the truer this will be, because of the "... geometric growth of interdependencies ... whose implications extend over time." (Pressman and Wildavsky [59]) Section 8 suggests some organizational mechanisms that can provide information systems developers with the authority and resources to resolve these complexities.

5. Counterimplementation

Believers in rationalism generally view resistance to change and protection of vested interests as faults to be ignored or suppressed. The tactical approach to implementation sees resistance as a signal from a system in equilibrium that the costs of change are perceived as greater than the likely benefits. The bringers and sellers of change—academics, computer specialists, and consultants—assume that what they offer is good. In practice, there are many valid reasons to go beyond passive resistance and actively try to prevent implementation. Many innovations are dumb ideas. Others threaten the interests of individuals and groups by intruding on their territory, limiting their autonomy, reducing their influ-

⁶ [73], p. 17.

ence, or adding to their workload. While we all may try to act in the "corporate" interest, we often have very different definitions of exactly what that is. (Dearborn and Simon point out that even senior executives adopt the perspective of their department [14]).

Obviously there is a fine line between honest resistance to a project one feels is misguided and selfish sabotage of a necessary innovation. The difference is a matter for conscience and self-scrutiny. In both cases, the response is political, whether "clean" or "dirty" politics.

Bardach [5] defines implementation as a game and outlines some of the moves and countermoves by which actors: (1) divert resources from a project; (2) deflect its goals; (3) dissipate its energies. A central lesson to be learned from examples of successful counterimplementation is that there is no need to take the risky step of overtly opposing a project. The simplest approach is to rely on social inertia and use moves based on delay and tokenism. Technical outsiders should be kept outside and their lack of awareness of organizational issues encouraged. ("Why don't you build the model and we'll deal with the people issues later; there's no need to have these interminable meetings.") If more active counterimplementation is needed, one may exploit the difficulty of getting agreement among actors with different interests by enthusiastically saying, "Great idea—but let's do it properly!" adding more people to the game and making the objectives of the venture broader and more ambitious and consequently more contentious and harder to make operational.

This author has found examples of most of the tactics Bardach identifies, in an ongoing study of the implementation of information systems and models for educational policy analysis in state government. Before discussing them, it is important to examine what is perhaps the single most important cause of counterimplementation in information systems development—the politics of data.

The link between control over information and influence has often been noted. "Information is a resource that symbolizes status, enhances authority and shapes relationships." (Wildavsky [74]) "Information is an element of power." (Quoted in Greenberger et al. [24]) Computer systems often redistribute information, breaking up monopolies. Building a database then becomes a political move; sometimes it is equivalent to a declaration of war. The system designer needs to ask:

- (1) Who owns the data?
- (2) Who will share it?
- (3) What will be the perceived impact of redistribution on:
 - (a) evaluation;
 - (b) influence and authority;
 - (c) communication?

He or she should then get ready to deal with counterimplementation.

Dornbusch and Scott define evaluation as central to the exercise of authority [18]. In general, providing management (or outside agencies) with data that permits closer observation of subordinates' decision making or helps define additional output measures increases control and decreases autonomy. Many public sector agencies protect data on their operations as a means of maintaining their independence. Laudon's study of information systems in local government provides many illustrations of this point [43], i.e., police agencies protect their data from mayors and budget agencies. *Information is control.* (See also Pettigrew [58].)

Evaluation and monitoring are often "improved" (from the manager's viewpoint) through the collection of routine operational data. An unanticipated side effect of information systems is an increase in the superior's ability to evaluate personnel. For example, telecommunications, office automation, and integrated databases provide and record simple access to information that may then be used to observe subordinates. The introduction of office automation has, for instance, led some managers to study "productivity" of clerical staff, measured in terms of lines typed or error rates. Hospitals similarly use computer-derived data to track nurses' performance; previously evaluation required interaction, some degree of negotiation and respect for the nurses' "professional" judgement. Some managers are concerned that trends in computer networking and database administration may similarly encourage their superiors to snoop.⁷

The link between evaluation and authority is recognized by many trade union leaders. Greenberger et al.'s discussion of the joint effort of Rand and the administration of Mayor Lindsay in New York to apply management science to city government provides several examples of their refusal to permit data to be gathered that might *later* be used to evaluate productivity [24]. Teacher unions similarly opposed efforts to introduce accountability programs. In at least one state, the Department of Education joined them in an elegant counterimplementation move, a variant of one Bardach [5] labels *Pile On*. Teacher accountability measures had been tacked onto a school finance bill. The Department of Education suggested six comprehensive programs, all of which involved collecting and processing additional data. It then scheduled about 30 statewide meetings, open to parents, the press, school officials, and teachers and loftily entitled "The Search for Consensus". This generated 44 separate accountability measures. The program is, of course, now dead. This counterimplementation was overt and skilled, but puzzling to analysts who saw the need for "better" data as in the interests of all.

A corollary of the link between evaluation and authority is the relationship between ownership of information and autonomy. In some cases, departments or individuals have influence only because they have a data

⁷ See [40]. [6] provides an excellent summary of power issues in relation to information systems, viewed mainly in terms of the accounting function.

Table I. Implementation Games (Bardach).

Diverting Resources	Sample Motivation
<i>Easy Money</i>	"Get a little more than we give back."
<i>Budget</i>	"We never turn down money."
<i>Easy Life</i>	"Make sure we're in charge and don't let outsiders cause trouble; take it slowly."
<i>Pork Barrel</i>	The elected official's version of Easy Money; "grab it while you can."
Deflecting Goals	
<i>Pile On</i>	"Let's do it right!—We have to make sure our interests are included in the project."
<i>Up for Grabs</i>	"If they don't know what they want, we'll take over."
<i>Keep the Peace</i>	"We're going to have to work closely with Marketing and make sure we're both happy."
Dissipating Energies	
<i>Tenacity</i>	"No." "One more time." "We're not happy about . . ."
<i>Territory</i>	"This is <i>our</i> job." "We think we should run the project since . . ."
<i>Not Our Problem</i>	"Marketing really ought to handle this."
<i>Odd Man Out</i>	"We're certainly interested and we'll be happy to provide some inputs, but . . ."
<i>Reputation</i>	"I want an integrated-on-line-real-time-database-management-distributed-processing-planning system. My system will . . ."

monopoly. (Cyert and March [13] comment that organizations are partly designed in terms of rules for filtering and channelling data. Particular units are given responsibility for collecting and interpreting data and other units may not challenge them.) Finance and Planning, for example, may own data on capital allocations. In state government agencies, budget officials often have a monopoly on the details of particular programs and expenditures which gives them great influence on the decision making process. Staff specialists, who often lack direct authority, rely on careful rationing of technical information in negotiations and on their ability to withhold data (Pettigrew [58]).

Information systems redistribute data and are sometimes intended to break up monopolies. This may be equivalent to redesigning parts of the organization, disrupting patterns of communication, and reallocating authority. Of course, this also means that they may be explicitly used to "... perpetuate or modify decision processes and social structures." (Bariff and Galbraith [6]) Information systems then become a tool for organizational development in the most literal sense of the term. The key point is that designers must recognize that far from being divorced from messy "politics", information technology has a major impact on a critical resource and source of power. It is hardly surprising then that teachers view a productivity reporting system as an outrage or that operating divisions oppose the efforts of Finance to coordinate planning through a budget tracking system. Computer specialists tend to be very surprised.

6. The Tactics of Counterimplementation

A key step in the tactical approach to implementation is to convert the general impetus for change which is

usually based on broad goals and rallying cries, into operational objectives and a specific contract. (Kolb and Frohman [41], Ginzberg [22]) Any project is very vulnerable to counterimplementation until this is done. Programs that have unclear goals or ambiguous specifications and that rely on continuing high levels of competence and coordination are easy targets for skilled game players. Bardach [5] outlines a variety of games. (Table I) *Easy Money* involves supporting a project because it can be used to finance some needed activity within the player's sphere of interest. The *Budget* game is played by managers as budget maximizers and *Territory* is similarly used to protect or extend control.

Within a game, there are some predictable moves. *Tenacity* exploits social inertia and interdependencies: "... all it takes is the ability and the will to stymie the completion, or even the progress, of a program until one's own particular terms are satisfied."⁸ *Odd Man Out* creates an option to withdraw if the project gets into trouble and then the chance to say "I told you so." This move is made easiest in projects where only the designer is accountable and no visible commitment is required from the game player. *Up For Grabs* is used to take over a program where the mandate is half-hearted or ambiguous.

All these moves are found in information systems development. There is an additional maneuver employed wherever computers are found—the *Reputation* game. Here, a manager gets credit as a bold innovator by sponsoring a new system—the closer to the state-of-the-art the better, since this increases his or her visibility and creates excitement. The *Reputation* gamer will have been transferred to a new position by the time the project collapses and can then ruefully say "... when I was in charge of things..." The short tenure of upwardly mobile managers and their need to produce fast results

⁸ [5], p. 148.

Table II. Scenario-Writing (adapted from Bardach).

A. Basic Objectives:	<ul style="list-style-type: none"> —What exactly are you trying to get done? (not what does the system look like?) —What resources are needed? —Who controls them, directly or indirectly? —How can you minimize the effects of social inertia?
B. Dilemmas of Administration:	<ul style="list-style-type: none"> —What elements are critical? —Are any of them subject to monopoly interests? —Will their owners be uncooperative? —Can you work around them or buy them off? —Will they respond with delays or tokenism? —How will you deal with massive resistance?
C. Games:	<ul style="list-style-type: none"> —What games are likely to <ul style="list-style-type: none"> (a) divert resources? (b) deflect goals? (c) dissipate energies? —How can you counteract or prevent them, if necessary by redesigning the project?
D. Delay:	<ul style="list-style-type: none"> —How much delay should you expect? —What negotiations are needed? —What resources do you have for negotiations and/or control? —Would it help to use project management, work around possible obstacles and delay or enlist intermediaries?
E. Fixing the Game:	<ul style="list-style-type: none"> —What senior management and staff aid do you need? —What resources do they have? —What incentives are there for them to play the fixer role? —Can you build a coalition to fix the game?

encourages this move, which is only possible however when the goals of the project are not made operational or specific commitments made to deliver phased outputs.

This analysis of implementation as a game may seem overcynical. However, it seems essential to ask at the start of a project:

- (1) Are people likely to play games?
- (2) Is the proposal proof against subversion?

These two simple questions provide the basis for a defensive strategy.

7. Counterimplementation: The Management Game

Most of the moves Bardach discusses exploit ambiguity and a lack of control mechanisms. The *Reputation* game player can get early credit and not be held accountable later. *Easy Money* is possible only because the goals of the project are too broadly stated. *Odd Man Out* occurs when technicians have to carry the venture (or choose to do so). Bardach suggests designers use "scenario-writing" (Table II) and in essence ask "who can

foul it up." The tactical approach to implementation makes the same recommendation, though more optimistically. At the Entry stage the implementer tries to identify and bring into the (facilitative) negotiations any party whose actions or inactions can affect the chances of success. Scenario-writing forewarns the designer and partially protects him or her against (1) monopoly and tokenism; (2) massive resistance; and (3) delays, deliberate or accidental. Bardach recommends a variety of responses to counterimplementation such as creating substitute monopolies (information systems personnel can use their specialized technical resources in this way for bargaining), co-opting likely opposition early, providing clear incentives ("If policy analysts carry bumper stickers, they should read 'Be Simple! Be Direct' or 'PAYMENT ON PERFORMANCE',⁹ (Pressman and Wildavsky [59]) and creating a bandwagon.

The Management game uses control mechanisms overlaid on others' games. By assigning priorities, developing project management procedures and above all, by keeping the scope of the project small and simple, which is often intellectually harder than designing a complicated system, the implementer can limit the range of moves actors can make. The Management game is difficult to play without a "fixer",¹⁰ a person or group with the prestige, visibility, and legitimacy to facilitate, deter, bargain, and negotiate effectively. Information systems teams often lack this key support.

8. Conclusion: A Strategic Perspective on Change

Counterimplementation (CCI) is largely defensive, whereas the facilitative tactical approach is proactive. To an extent, CCI involves containing and doing the opposite of counterimplementers, whose strategy may be summarized as:

- (1) Lay low;
- (2) Rely on inertia;
- (3) Keep the project complex, hard to coordinate, and vaguely defined;
- (4) Minimize the implementers' legitimacy and influence;
- (5) Exploit their lack of inside knowledge.

The tactical model addresses some of these issues:

- (1) Make sure you have a contract for change;
- (2) Seek out resistance and treat it as a signal to be responded to;
- (3) Rely on face-to-face contracts;
- (4) Become an insider and work hard to build personal credibility;
- (5) Co-opt users early.

⁹ [59], p. 159.

¹⁰ [5], pp. 273–278. The concept of a fixer vastly extends the platitude in the implementation literature of the need for top management support.

A strategic model for change needs to resolve some additional concerns:

- (1) What happens when consensus is impossible?
- (2) How can large-scale projects evade social inertia?
- (3) What authority mechanisms and organizational resources are needed to deal with the politics and data and counterimplementation?
- (4) What is the role of management?

Some points are obvious from the analysis so far. Whether we like it or not, we can only hope for incremental change [except, as Ansoff points out [2] in situations of mild crises, where the status quo is no longer satisfactory, and organizations rethink their goals and are more willing to think "rationally"]. This reality suggests that systems designers must always aim for simplicity of design and precise objectives. However, if they are to go beyond tactical innovations based on Up-and-In, they need Down-and-Out directional planning; they must establish the *direction* of change and evolve complex systems out of phased components. This requires nontechnical resources such as (1) a meaningful steering committee and (2) authority.

The analysis in this paper indicates that information development must be spearheaded by a general, not coordinated by aides-de-camp. It must be defined as part of the Information function of the organization, instead of being a staff service labelled data processing or management science. The issues of negotiations seem central. (Kling and Gerson [40]) To position a system one must clarify objectives, respond to resistance, adjust other components of the Leavitt Diamond (**Task, Technology, People, Structure**) and block off counterimplementation. The politics of data (and of software engineering; see Keen and Gerson [35]) make it essential that negotiations be handled by a fixer, well-linked into senior managements' decision making. Large scale change is a process of coalition-building; this cannot be done by staff analysts, who are too easily caught in the middle with no formal powers.

The strategy for managing social change is based on acceptance of the political nature of information systems development and the need for suitable authority. Many organizations have moved in this direction. Neal and Radnor and their colleagues [56, 62] conclude that OR/MS groups with formal charters (budgets, senior job titles for their managers, and the right to turn down user requests) are more successful than ones that are a corporate service unit. The few Grand Old Men in the information systems field who have risen to senior positions in large companies have built up organizational mechanisms that provide them with authority and strong links with top level planning in the organization. (Strassman [66], Edelman [20]) There is perhaps an almost Darwinian process of natural selection, where the MIS group adopts a purely technical focus or cannot obtain authority for negotiations, it becomes merely a data processing service limited to routine applications and

subject to all the forces of inertia and counterimplementation discussed here.¹¹

It is not the aim of this paper to define a specific strategy for implementation. The outline seems clear:

- (1) A senior level fixer must head the Information function; he or she must have full authority and resources to negotiate with or between users and with those affected by information systems;
- (2) There must be some policy planning or steering committee which includes senior line managers; it will delegate to technical staff responsibility for projects that do not have significant organizational impact but will be actively involved with ones that are part of the politics of data (the policy committee also provides a negotiating table);
- (3) The planning process will require substantial time and effort in the *predesign* stages, where objectives are made operational and evolution of the larger system is defined by breaking it into clear phases;
- (4) Formal contracts will be needed, in which commitments must be clearly made and such games as *Up for Grabs, Reputation, Easy Life, and Territory* made illegal and ineffectual;
- (5) "Hybrid" skills must be developed in systems staff; they cannot dismiss organizational and political issues as irrelevant or not their responsibility, but must be able to operate in the manager's world and build credibility across the organization.¹²
- (6) With the umbrella provided by the fixer's authority and the steering committee, the tactical approach remains an excellent guide to managing the implementation process for a given project.

The simple, central argument presented here is that information systems development is political as well as, sometimes far more so than, technical in nature. When that is accepted, the organizational mechanisms follow naturally. Unfortunately, "politics" have been equated with evil, corruption and, worst of all, blasphemy in the presence of the Rational Ideal, but politics are the process of getting commitment, or building support, or creating momentum for change; they are inevitable.

The final comments to be made here concern research. There have been few studies of the political aspects of information systems development. The topic is rarely discussed in textbooks and even the literature on tactical implementation deals with it only peripherally. Yet when one tries to reconstruct or observe the progress of any major project, this is an obvious and important feature. It is absurd to ignore it or treat it as somehow an unsuitable subject for study or for training MIS specialists. There is some fragmented research available: Pettigrew's observation of a computer purchase decision, [57] Laudon's, *Computers and Bureaucratic Re-*

¹¹ See [35] and [32], who argues that most MIS groups are locked into the "maintenance" activities of the organization which reinforces the status quo and emphasizes efficiency. They have little impact on the "adaption" functions, which involve innovation and strategic planning.

¹² See [36], Chap. 9.

form. [43] and the work done by the Urban Information Systems Research Group at the University of California at Irving. (Kling [38], [39]) Greenberger et al. [24] also provide some vivid illustrations of the political nature of computer models in public policy making. Most of this work is based on case studies. Politics are hard to study. They involve many hidden agenda (counterimplementers do not boast about their triumphs) and in most instances a skilled observer has to ferret out and interpret what has happened. In political science, the work on implementation is almost entirely narrative and descriptive. A political perspective on information systems is needed in research. It will of necessity be based on comparative field studies that illustrate theoretical concepts.¹³ It will not fit the standard mold for behavioral research. It can immensely add to our understanding both of the implications of information technology and the dynamics of effective implementation. For a long time the word "implementation" was not included in the index to literature on OR/MS and MIS. It is to be hoped that "politics", "negotiations", and "authority" be increasingly found in the titles of papers on information systems. That the papers will often be case studies does not mean they are not "legitimate" research. We badly need more understanding of these issues which are of fundamental importance to the effective exploitation of computer technology.

Received 3/79; revised 5/80; accepted 10/80

¹³ Mintzberg [52] provides a full discussion of the difficulties of studying phenomena which involve "soft" variables and need an integrating perspective. His own field research [51] is a striking example of how much we can learn from simple, imaginative observation, which often conflicts with complex over-narrow experimentation.

References

Reference [53] is not cited in text.

1. Allison, G.T. *Essence of Decision*. Little Brown, Boston, Mass., 1971.
2. Ansoff, H.I., *Business Strategy*. Penguin, London, England, 1968.
3. Argyris, C. Resistance to rational management systems. *Innovation*, 10, (1970) pp. 28-35.
4. Argyris, C. Management information systems: The challenge to rationality and emotionality. *Management Sci.* 17, 6, (1971), pp. B275-292.
5. Bardach, E. *The Implementation Game: What Happens After a Bill Becomes a Law*. MIT Press, Cambridge, Mass., 1977.
6. Bariff, M.L. and Galbraith, J.R. Intraorganizational power considerations for designing information systems. The Wharton School, Univ. of Penn., Philadelphia, Penn., Jan. 1978.
7. Bower, J. *The Resource Allocation Process*. Irwin, New York, 1970.
8. Bowman, E.H. Consistency and optimality in managerial decision making. *Management Sci.* 9, 2, (Jan. 1963) pp. 310-321.
9. Braybrooke, D. and Lindblom, C.E. *A Strategy of Decision*. Free Press, New York, 1963.
10. Chesler, M. and Flanders, M. Resistance to research and research utilization: The death and life of a feedback attempt. *J. Appl. Behavioral Sci.* 3, 4, (Fall 1967), pp. 465-487.
11. Churchman, C.W. Managerial acceptance of scientific recommendations. *Calif. Management Rev.* 7, 1 (Fall 1964), pp. 31-38.
12. Cohen, M.R. and March, J.G. *Leadership and Ambiguity*. McGraw-Hill, New York, 1974.
13. Cyert, R.M. and March, J.G. *A Behavioral Theory of the Firm*. Prentice-Hall, Englewood Cliffs, N. J., 1963.
14. Dearborn, D.C. and Simon, H.A. The identification of executives. In *Administrative Behavior* (3rd ed.). H.A. Simon, Ed. Free Press, New York, 1976, pp. 309-314.
15. Derthick, M. *New towns in-town*. The Urban Institute, Washington, D. C., 1972.
16. Doktor, R.H. Development and mapping of certain cognitive styles of problem-solving. Unpublished Ph.D. Dissertation, Stanford Univ., Stanford, Calif., 1969.
17. Doktor, R.H. and Hamilton, W.F. Cognitive style and the acceptance of management science recommendations. *Management Sci.* 19, 8 (April 1973), pp. 884-894.
18. Dornbusch, S. and Scott, W.R. *Evaluation and the Exercise Authority*. Jossey-Bass, San Francisco, Calif., 1975.
19. Drake, J.W. *The Administration of Transportation Modelling Projects*. Heath, Lexington, Mass., 1972, pp. 14-17.
20. Edelman, F. Four ways to oblivion—A short course on survival. *Interfaces* August, 1972, pp. 14-17.
21. Galbraith, J.R. *Designing Organizations*. Addison-Wesley, Reading, Mass., 1977.
22. Ginzberg, M.J. A process approach to management science implementation. Unpublished Ph.D. Dissertation, Sloan School of Management, M.I.T., Cambridge, Mass., 1975.
23. Grayson, C.J. Management science and business practice. *Harvard Business Rev.* 51, 4 (July-Aug. 1973), pp. 41-48.
24. Greenberger, M., Crenson, M.A., and Crissey, B.L. *Models in the Policy Process*, Russell Sage Foundation, New York, 1976.
25. Hall, W.K. Strategic planning models: Are top managers really finding them useful? *J. Business Policy* 3, 3, 1973, pp. 19-27.
26. Hall, W.K. Rationality, irrationality and the policy formulation process in large organizations. *Planning Rev.* 4, 6 (May 1976), pp. 22-26.
27. Hargrove, E.C. The missing link: The story of implementation process. The Urban Institute, Washington, D. C., 1975.
28. Hirschman, A.O. *The Strategy of Economic Development*. Yale Univ. Press, New Haven, Connecticut, 1958.
29. Hoos, I.R. *Systems Analysis in Public Policy*, Univ. of Calif. Press, Berkeley, Calif., 1972.
30. Huysmans, H.B.M. *The Implementation of Operations Research*. Wiley, New York, 1970.
31. Kahneman, D. and Tversky, A. Prospect theory: An analysis of decision under risk. *Econometrica* 47, (March 1979) p. 263.
32. Keen, P.G.W. Managing organizational change: The role of MIS. In *Proc. 6th and 7th Ann. Conf. of the Soc. for Management Infor. Syst.*, J.D. White, Ed. Univ. of Michigan, Ann Arbor, Mich., July 1976, pp. 129-134.
33. Keen, P.G.W. The evolving concept of optimality. In *Multi-Criteria Decision Making*. M.K. Starr and M. Zeleny, Eds., The Inst. of Management Sci. (TIMS), Studies in Management Sci., 6, 1977, pp. 31-57.
34. Keen, P.G.W. Implementation research in MIS and OR/MS: Description versus prescription. Stanford Business School Research Paper No. 390, Stanford, Calif., 1977.
35. Keen, P.G.W. and Gerson, E.M. The politics of software engineering. *Datamation* 23, 11 (Nov. 1977), pp. 80-86.
36. Keen, P.G.W. and Scott Morton, M.S. *Decision Support Systems: An Organizational Perspective*. Addison-Wesley, Reading, Mass., 1978.
37. Klein, B. and Meckling, W. Applications of operations research to development decisions. *Operations Res.* 6, 3, (May-June 1958), pp. 352-363.
38. Kling, R. Information systems in policy making. *Telecomm Policy*, 2, 1, (March 1978), pp. 3-12.
39. Kling, R. Social analyses of computing: Theoretical perspectives in recent empirical research. *Computing Survey* 12, 1 (March 1980) pp. 61-110.
40. Kling, R. and Gerson, E.M. The social dynamics of technical innovation in the computing world. *Symbolic Interaction*, 1, 1 (Fall 1977) pp. 132-146.
41. Kolb, D.A. and Frohman, A.L. An organizational development approach to consulting. *Sloan Management Rev.* 12, 1 (Fall 1970) pp. 51-65.
42. Kunrueher, H. and Slovic, P. Economics, psychology and protective behavior. *Am. Econ. Rev. Papers and Proceedings*. May 1978.
43. Laudon, K.C. *Computers and Bureaucratic Reform: The Political Functions of Urban Information Systems*. Wiley, New York, 1974.

44. Leavitt, H.J. Applying organizational change in industry: Structural, technological and humanistic approaches. *Handbook of Organizations*, J.G. March, Ed. Rand McNally, Chicago, Ill., 1965.
45. Leavitt, H.J. and Webb, E. Implementing: Two approaches. Stanford Univ. Research Paper 440, Stanford, Calif., May 1978.
46. Lindblom, C.E. The science of muddling through. *Public Administration Rev.*, 19, 2, (Spring 1959), pp. 79-88.
47. Lindblom, C.E. *Politics and Markets*. Basic Books, New York, 1977.
48. Lucas, H.C. and Plimpton, R.B. Technological consulting in a grass roots action-oriented organization. *Sloan Management Rev.* 14, 1 (Fall 1972) pp. 17-36.
49. McKenney, J.L. and Keen, P.G.W. How managers' minds work. *Harvard Business Rev.* 52, 3 (May-June 1974) pp. 79-90.
50. Miller, R.B. Psychology for a man-machine problem-solving process. IBM Data Systems Division Laboratory, Rept TR00-1246, February 1965.
51. Mintzberg, H. *The Nature of Managerial Work*. Harper and Row, New York, 1973.
52. Mintzberg, H. Policy as a field of management theory. *Academy of Management Review*, January 1977. (Unnumbered paper).
53. Mintzberg, H. Beyond implementation: An analysis of the resistance to policy analysis. *Infor* 18, 2 (May 1980) pp. 100-138.
54. Mitroff, I.I. *The Subjective Side of Science: A Philosophic Enquiry into the Psychology of the Apollo Moon Scientists*. Elsevier, New York, 1974.
55. Munson, F.C. and Hancock, W.M. Problems of implementing change in two hospital settings. *AIIE Trans.* 4, 4 (Dec. 1977) pp. 258-266.
56. Neal, R.D. and Radnor, M. The relation between formal procedures for pursuing OR/MS activities and OR/MS group success. *Operations Res.* 21, (March 1973), pp. 451-474.
57. Pettigrew, A.M. *The Politics of Organizational Decision Making*. Tavistock, London, England, 1973.
58. Pettigrew, A.M. Implementation control as a power resource. *Sociology* 6, 2, (May 1972) pp. 187-204.
59. Pressman, J. L. and Wildavsky, A. *Implementations*. Univ. of Calif. Press, Berkeley, California, 1973.
60. Roberts, E.B. On implementing systems studies: Strategies for achieving organizational change in response to model-based analysis. Paper presented at the American-Soviet Conference Series on Methodological Aspects of Social Systems Simulation, Sukhumki, USSR, Oct. 17-23, 1973.
61. Ross, L. The intuitive psychologist and his shortcomings in *Advances in Experimental Social Psychology*. L. Berkowitz Ed. Academic, New York, 1964, pp. 173-220.
62. Rubenstein, A.H., Radnor, M., Baker, N., Heiman, D. and McCoy, J. Some organizational factors relative to the effectiveness of management science groups in industry. *Management Sci.* 13, 8, (April, 1967), pp. B508-518.
63. Saposky, H.M. *The Polaris System Development*. Harvard Univ. Press, Cambridge, Mass. 1972.
64. Simon, H.A. A behavioral model of rational choice. In *Models of Man*, H.A. Simon, Ed. Wiley, New York, 1957, pp. 241-260.
65. Stewart, R. *Managers and Their Jobs*. McMillan, London, England, 1967.
66. Strassman, P. Managing the costs of information. *Harvard Business Rev.*, 54, 5 (Sept-Oct 1976), pp. 133-142.
67. Strauss, A. *Negotiations: Varieties, Contexts, Processes, and Social Cries*. Jossey-Bass, San Francisco, Calif., 1978.
68. Tversky, D. and Kahneman, A. Judgment under uncertainty: Heuristics and biases. *Science*, 185, (Sept. 1974), pp. 1124-1131.
69. Urban, G.L. Building models for decision makers. *Interfaces* 4, 3, (May 1974) pp. 1-11.
70. Vancil, R. and Lorange, P. *Strategic Planning Systems*. Prentice-Hall, Englewood Cliffs, N. J., 1977.
71. Vertinsky, I.R., Barth, T., and Mitchell, V.F. A study of OR/MS implementation as a social change process in *Implementing Operations Research/Management Science*, R.L. Schultz and D.P. Slevin, Eds. American Elsevier, New York, 1975, pp. 253-272.
72. Weick, K. *The Social Psychology of Organizing*. Addison-Wesley, Reading, Mass., 1969.
73. Weiner, S. and Wildavsky, A. The prophylactic presidency. *Public Interest* 52, 52, (Summer 1978), pp. 3-19.
74. Wildavsky, A. *The Politics of the Budgetary Process*. (2nd Ed.), Little, Brown, Boston, Mass., 1974.
75. Wilensky, H.L. *Organizational Intelligence: Knowledge and Policy in Government and Industry*. Basic Books, New York, 1967.
76. Zand, D.E. and Sorenson, R.E. Theory of change and the effective use of management science. *Administrative Sci Quarterly*, 20, 4, (Dec. 1975), pp. 532-545.
77. Zionts, S. and Wallenius, J. An interactive programming method for solving the multiple criteria problem. *Management Sci.* 22, 6, (Feb. 1976), pp. 652-663.

23 Valkuilen bij automatisatieprojecten

door ir. S. M. Argelo

Dit artikel vestigt de aandacht op vermijdbare fouten die kunnen worden gemaakt bij de ontwikkel- en implementatie fasen van een bedrijfsinformatiesysteem. Er wordt gepleit voor een evenwichtige behandeling van het automatisatieproject, zodat naast technische en economische ook sociaal-organisatorische criteria voldoende aandacht krijgen.

De mogelijk optredende problemen worden geïdentificeerd als stoornissen in de relaties tussen betrokken partijen; te weten gebruikers, systeemontwerpers en managers. Bovendien worden de problemen volgordelijk behandeld, in relatie tot de project life cycle. Het artikel mondt uit in een aantal aanbevelingen, gericht tot elk van de genoemde partijen.

1 INLEIDING

Dit artikel gaat over automatisatieprojecten en hun kans van slagen. Meer in het bijzonder hebben de auteurs het oog op het soort project dat uitmondt in een bedrijfsinformatiesysteem. Zo'n informatiesysteem ontleent zijn waarde in de eerste plaats aan de bijdrage die het levert als hulpmiddel bij besluitvorming en bedrijfsvoering in een organisatie.

De auteurs menen uit eigen ervaring te kunnen stellen dat in veel gevallen de beloften niet of niet ten volle worden waargemaakt. Automatisatieprojecten duren veelal langer dan was gepland, kosten meer dan was gebudgetteerd en voldoen matig tot slecht aan de wensen van de gebruikers. Het vermoeden bestaat dat organisatorische en sociale overwegingen naar verhouding onvoldoende aandacht krijgen. Een relatieve overwaarding van technische en economische aspecten tijdens ontwikkeling en implementatie zou wel eens bedreigend kunnen zijn voor de effectiviteit van informatiesystemen.

Om verwarring te voorkomen is het goed om enkele sleutelbegrippen te verduidelijken. Het gaat hier om de begrippen

- (automatie-)project
- informatiesysteem
- effectiviteit
- efficiency

Een *project* is een verzameling onderling samenhangende activiteiten, begrensd door een doelstelling, een budget en een tijdsplanning.

Een *automatieproject* heeft tot doel een informatiesysteem tot stand te brengen en in een organisatie te implementeren; in het algemeen speelt elektronische gegevensverwerking daarbij een belangrijke rol.

Een *informatiesysteem* is een werkend geheel van apparatuur, programmatuur, procedures en mensen; mensen zijn een essentieel deel van het systeem omdat:

- mensen het systeem met invoer moeten voeden

- mensen met de uitvoer verder moeten werken
- mensen het systeem moeten beheren als een dynamisch element in hun organisatie.

Bij het ontwikkelen, bouwen en implementeren van informatiesystemen zijn effectiviteit en efficiency sleutelwoorden. Onder *effectiviteit* verstaan we de mate waarin het gestelde doel bereikt wordt als het informatiesysteem operationeel is. Met *efficiency* bedoelen we een integrale kosten/baten afweging. Integraal omdat we niet alleen moeten kijken naar technische- en economische aspecten maar ook naar sociale en organisatorische aspecten, zowel kwantitatief als kwalitatief.

Effectiviteit en efficiency worden beïnvloed door een veelheid van factoren. Door de voortdurend goedkoper wordende apparatuur voor informatieverwerking, en de daardoor gestimuleerde tendensen tot organisatorische decentralisatie, wordt de invloed van eindgebruikers op de uiteindelijke effectiviteit en efficiency van informatiesystemen van steeds groter gewicht.

De wijze en het tijdstip waarop eindgebruikers betrokken worden in bouw, ontwikkeling en implementatie heeft grote invloed op de mate van acceptatie en dus ook op de effectiviteit van het nieuwe hulpmiddel.

2 PARTIJEN OF PARTNERS

Om tot beter begrip te komen met betrekking tot de kans van slagen, is het nodig het automatisatieproject te bekijken in relatie tot zijn omgeving. Op een automatisatieproject wordt vanuit de organisatie invloed uitgeoefend van velerlei aard. Die invloedssferen bepalen de inputs op het project, maar verduidelijken ook waar de consequenties van het project worden gevoeld. Het gaat voornamelijk om invloeden en consequenties van de volgende aard:

- informatie-technische
 - economische
 - organisatorische
 - sociale
-

Deze vier invloedssferen vallen in feite samen met de belangrijkste aspecten van het bedrijfsgebeuren, en dat is niet verwonderlijk, want aan de bijdrage tot de effectiviteit van het bedrijfsgebeuren dienen de uitkomsten van het project worden getoetst. De resultaten van een automatisatieproject te moeten aan de eis voldoen dat het functioneren van de organisatie in één of meer aspecten erdoor verbetert, en in zijn overige aspecten op zijn minst niet verslechtert.

We kunnen het automatisatieproject ook in verband brengen met een drietal in de organisatie herkenbare instanties, die tevens fungeren als capaciteitsbronnen.

G = de *Gebruikers* van het op te leveren informatiesysteem; bij hen overwegen invloeden en consequenties van sociale en organisatorische aard.

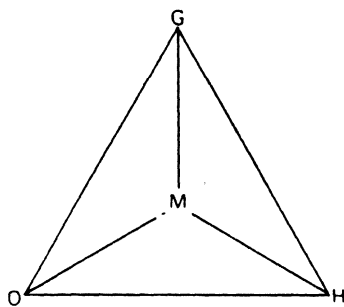
O = de *Ontwerpers* van het systeem, in hoofdzaak zijn dat systeemanalisten en programmeurs; bij hen is de informatie-technische inbreng dominant.

M = het *Management*, overkoepelend ten opzichte van Gebruikers en Ontwerpers, met accent op economische en organisatorische aspecten.

Aan dit drietal moet nog een vierde element worden toegevoegd, om het beeld te completeren. Dat vierde structurelement, aan te duiden met de letter H van Hulpbronnen, omvat alle relevante capaciteiten welke niet in de trits G/O/M zijn begrepen, zoals

- reeds eerder geïmplementeerde informatiesystemen
- apparatuur en basis-programmatuur
- edukatieve en procesbegeleidende deskundigheid
- financieel-economische deskundigheid
- procedures en standaards

De vier op het automatisatieproject inwerkende structurelementen willen we rangschikken in een diagram, waarin hun onderlinge relaties worden gesymboliseerd.



De Gebruikers van het systeem zijn in de top gesitueerd. Hun problemen, behoeften en wensen dienen het vertrekpunt te zijn voor het automatisatieproject; van hun acceptatie en satisfactie zal afhangen in hoeverre het op te leveren systeem effectief zal zijn. De basis van de driehoek wordt gevormd door Ontwerpers en (andere) Hulpbronnen; de bijdragen van deze beide structurelementen zijn meer instrumenteel en minder belanghebbend dan in het geval van de Gebruikers. Centraal staat het Management, vooral te zien als initiërende en coördinerende intermediair voor de overige drie.

Tussen elk paar van deze vier bestaan relaties. De knelpunten in elk van de zes relaties zijn nader uitgewerkt in hoofdstuk 4.

Vooruitlopend op onze konklusies kan nu al worden opgemerkt dat de kans van slagen van een automatisatieproject (en dus de effectiviteit van het te implementeren informatieverwerkende systeem) afhangt van het gedrag van Gebruikers, Ontwerpers en Management in relatie tot elkaar en tot hun Hulpbronnen. Als ze zich als *partijen* opstellen, in een afgedwongen participatie, dan is de effectiviteit gering; als ze zich daarentegen als *partners* gedragen dan is de kans op succes groot.

3 PRODOSTA

Voor het beheersen van een automatisatieproject van enige omvang is het nodig te beschikken over een checklist van activiteiten en een documentatiestandaard. Een voorbeeld van zo'n methode is PARAET (zie INFORMATIE, jrg. 23 nr. 10 (oktober 1980)). Een ander voorbeeld is PRODOSTA, o.a. in gebruik bij DSM en Philips, en op de markt gebracht door VOLMAC. Verderop in dit artikel wordt enige bekendheid met PRODOSTA (Project Control and Documentation Standards) verondersteld, reden waarom hier kort wordt uiteengezet hoe het werkt.

PRODOSTA verdeelt de 'project life cycle' in 6 fasen, elk van die fasen wordt ondersteund door een gedetailleerde activiteitenlijst. Een fase wordt afgesloten met een 'Baseline document', waarop autorisatie nodig is alvorens de volgende fase kan worden ingegaan. Elke baseline wordt ondersteund door een model inhoudsopgave van de te schrijven documentatie.

	FASERING	BASELINES
1	Vooronderzoek	Application request
2	Toepassingsanalyse	Application proposal
3	Systeemspecificatie	System requirements specification
4	Systeemdefiniëring	System design specification
5	Systeembouw	System manuals
6	Operationele fase	Performance evaluation report

Voor een gedetailleerd overzicht van de werkzaamheden per fase wordt verwezen naar de PRODOSTA documentatie. Binnen het kader van dit artikel is het voldoende om kennis te nemen van de hier volgende korte weergave van de inhoud van elk van de afsluitende baselinedocumenten. Daarmee is immers ook in grote lijnen bepaald welke acties eraan voorafgaan.

- 1 Het 'Application request' is een opdracht van het bevoegde management om te starten met de projectaanpak van een geïdentificeerd informatieprobleem, dat is voortgekomen uit een bedrijfsanalyse.
- 2 Het 'Application proposal' bevat het resultaat van een toepasbaarheidsonderzoek en een uit-

- gewerkt plan voor een projekt. Dat projekt zal leiden tot een informatiesysteem met bekende doelstellingen, waaraan het resultaat kan worden getoetst.
- 3 De 'System requirements specification' geeft het resultaat van het functioneel ontwerp. Het is de 'uitwendige' beschrijving van het te bouwen systeem, in termen van beheersprocedures, I/O-procedures, scherm layouts, betrouwbaarheid, flexibiliteit, responsiesnelheid enz.
 - 4 De 'System design specification' geeft het resultaat van het technisch ontwerp. Het is de 'inwendige' beschrijving van het systeem, te weten hardware-komponenten, programma's, bestanden, kommunikatieprocedures enz.
 - 5 De 'System manuals' bevatten voor elk van de betrokken afdelingen (gebruikers, beheerders, onderhoudsmensen) de definitieve informatie benodigd voor gebruik en onderhoud. Als er tijdens de programmering en implementatie (dus in fase 5) geen gekke dingen gebeurd zijn, dan vormen de baselines 3 en 4 een goede eerste versie van baseline 5.
 - 6 Een 'Performance evaluation report', dat periodiek kan verschijnen, is de afweging tussen doelstellingen enerzijds en ervaringen met het systeem anderzijds, en bevat leerzame conclusies voor de toekomst.

We zullen aannemelijk maken dat het beschikken over een methode als PRODOSTA geen garantie voor succes is. Het hangt er vanaf hoe het gereedschap gebruikt wordt. We zullen dan ook niet aandringen op het ontwikkelen van een nieuwe methodologie. Aan PRODOSTA mankeert niet veel, en met PARAET, Pandata's SDM of andere derivaten van ARDI en Systems Management is ook best te leven. Het gaat ons om de attitude van gebruikers, managers en ontwerpers ten opzichte van elkaar en van hun formele afspraken. We menen dat die attitude beïnvloedbaar is. Een geschikt moment daarvoor is tijdens fase 1, in een workshop met alle bij een concreet projekt betrokkenen, waarbij dit artikel leidraad voor de discussie zou kunnen zijn.

4 VERSTOORDE RELATIES

Op grond van ervaring hebben we een lijst opgesteld van circa 80 mogelijke knelpunten of belemmeringen die bij een automatieprojekt een rol kunnen spelen. Het bleek mogelijk deze knelpunten naar twee gezichtspunten in te delen, te weten:

- in de 6 relaties die tussen Gebruikers, Ontwerpers, Management en Hulpbronnen kunnen worden onderscheiden.
- in de tijd, door vast te stellen in welk van de 6 PRODOSTA fasen elk knelpunt van overwegend belang is.

De hiervoor genoemde lijst van knelpunten was voor ons zelf een onmisbaar hulpmiddel. Als onderdeel van deze publikatie leek zo'n dorre opsomming echter minder geschikt. We hebben daarom in dit hoofdstuk voor elk van de zes eerder gedefinieerde relaties geschetst waar het fout gaat. Alle mogelijke knelpunten

zijn daarbij verwerkt. Daardoor zijn deze zes schetsen gechargeerd. Alles wat mis kan gaan, loopt in deze zwartgallige verhaaltjes ook inderdaad uit de hand. De lezer zal dus hopelijk een groot aantal beweringen tegenkomen waarbij hij verontwaardigd (of vergenoegd) zal uitroepen: 'maar dát soort fouten maken wij nooit!' Daarnaast zullen er waarschijnlijk voor iedereen momenten zijn waarop, de hand oprecht in eigen boezem stekend, zal moeten worden erkend: 'Ja, in dát opzicht vliegen we ook meestal uit de bocht.' In de praktijk blijkt overigens dat slechts een deel van de gesignaleerde fouten nodig is om tot een ineffektief automatie-systeem te komen.

4.1 Mogelijke problemen in de relatie Gebruikers-Ontwerpers

De ontwerpers beschikken reeds bij aanvang van het projekt over een enorme kennisvoorsprong, vooral t.o.v. de eindgebruikers. Deze 'know-how gap' wordt zorgvuldig in stand gehouden en verder verdiept, door toepassing van technisch jargon. De ontwerpers zijn erop uit de nieuwste nieuwtjes op informatica-gebied toe te passen (technological push).

De gebruikers zijn verdeeld over diverse afdelingen en hiërarchische niveaus. Zij geven de ontwerpers alle kans om een verdeel- en heers-politiek te voeren. De werkervaringen op de vloer, broodnodig in de specificatie- en ontwerpfasen, worden niet gemobiliseerd. De gebruikers hebben daardoor in de implementatiefase nog ruim voldoende argumenten beschikbaar om de zaak op losse schroeven te zetten. Uit gemakzucht delegeren de gebruikers alle verantwoordelijkheid naar de specialisten, die intussen enthousiast bezig zijn allerlei zelfbedachte toeters en bellen aan het systeem te bouwen. Als er al PRODOSTA of een andere formele projektbesturingsmethode wordt toegepast, dan zorgen de ontwerpers er wel voor dat de baseline-dokumenten moeilijk toegankelijk zijn. Dat is trouwens nauwelijks nodig, want de toekomstige gebruikers zijn toch te lui om die dikke pakken papier te lezen. In extreme gevallen leidt dit er zelfs toe dat het systeem bij oplevering voor de gebruikers een volslagen verrassing is, en dan meestal geen aangename.

Aan beheersaspecten wordt nauwelijks aandacht besteed, want dat is taai, procedureel gedoe, waaraan ontwerpers een hekel hebben. Zij voeren zelf in de laatste fase vóór oplevering het systeembeheer op informele wijze, en daarin slagen ze door hun kennis van zaken. Aan produktievoorbereiding wordt weinig gedaan, zodat het systeem nauwelijks geschikt is om te worden beheerd door mensen die in het ontwerp niet thuis zijn en ook niet de opleiding hebben gehad om met het systeem om te gaan op de wijze zoals ontwerpers dat kunnen.

Aan het eind van de rit wordt het projekt gekapt. Ook is mogelijk dat een sterk management het er formeel doordrukt, waarna het systeem door de eindgebruikers niet of slechts in schijn wordt toegepast. Dubbele procedures blijven bestaan; men benut slechts een deel van het systeem, of nog erger: men probeert het systeem te verslaan. De ontwerpers vragen tevergeefs om discipline, en de gebruikers reageren met de klacht dat met het systeem niet te werken is.

4.2 Mogelijke problemen in de relatie Gebruikers-Hulpbronnen

Een instrument tot projektbeheersing, zoals PRODOSTA, werkt alleen als alle betrokken partijen (management, ontwerpers en gebruikers) er actief gebruik van maken. Het management wil dat nog wel eens doen, ziet in PRODOSTA een handig controle-middel (budget- en tijdbewaking). De ontwerpers hanteren PRODOSTA als een checklist die hen in-dekt tegen onvolledigheid.

De gebruikers zijn echter het minst geneigd om PRODOSTA te zien als een hulpmiddel van en voor hen-zelf. Het project is al goed en wel in fase 3 voordat de gebruikers zich hebben georganiseerd. De belangrijkste fasen 1 en 2 verlopen dus zonder gebruikersbemoediging. Van planning en voortgangskontrolle hebben de meeste gebruikers geen kaas gegeten. Ze laten dat over aan het management en aan de automatise-afdeling, maar ontwerpers plannen niet het gebruikersaandeel, en het management weet er inhoudelijk te weinig van.

De gebruikers zijn dus weinig of niet rechtstreeks bij het project betrokken. Wellicht zijn daarvoor excuses aan te voeren als ondeskundigheid en drempelvrees. Deze verontschuldigheden gelden niet voor de vele andere kansen op informatie en invloed die de gebruikers eveneens onbenut laten. De voor werkoverleg uitgetrokken tijd wordt geheel aan lopende zaken besteed, het project vormt nog geen aktueel probleem en blijft daarom buiten de gezichtskring. Ook de Ondernemingsraad speelt niet mee, want het automatise-projekt is door niemand gezien als organisatieverandering en dus ook niet als zodanig aangemeld. De roep om voorlichting, opleiding en demonstraties wordt pas gehoord als het te laat is om nog enige terugkoppeling op de systeemspecificaties te geven.

Als het systeem klaar is, en dus een hulpbron voor de organisatie zou moeten zijn, ontdekken de gebruikers dat er nogal wat discipline en taai volharding nodig is om de invoer telkens op tijd aan te leveren en het operationele beheer akkuraat uit te voeren. Ze voelen zich een verlengstuk van het systeem in plaats van andersom. Dat vinden ze niet leuk, het systeem werkt dus niet best en levert kwalitatief slechte uitvoer. Daarmee wordt het bewijs geleverd dat het zonder dat systeem veel beter ging, iets wat de gebruikers altijd al dachten.

4.3 Mogelijke problemen in de relatie Ontwerpers-Hulpbronnen

De ontwerpers zijn deskundig op hun terrein, dat wil zeggen systeemanalyse, programma-ontwerp, programmering en test. Dat er ook andere deskundigheden nodig zijn om het projekt te doen slagen is een lastige bijkomstigheid. Organisatiekunde, financieel-ekonomische expertise, sociale en edukatieve vaardigheden zijn onwelkome elementen op een werkterrein dat door de informatici eigenlijk wordt gezien als het domein van een op zichzelf staande technische ontwikkeling.

Ontwerpers willen ontwerpen, en hebben dus een grondige hekel aan standaard-oplossingen, applicatiepakketten e.d.; het programma dat ze in het

hoofd hebben is altijd mooier dan het programma dat wordt opgeleverd, en dat geldt zeker voor het slecht onderhoudbare, onduidelijk gedokumenteerde brodelwerk van iemand anders.

De automatise-groep werkt in een nauwkeurig afgesloten keuken, met apparatuur en programmatuur waarvan zij alleen de geheimen kennen. Het in aanbouw zijnde applicatiesysteem moet helemaal af vóór het aan het volk kan worden getoond. Er wordt dus geen gebruik gemaakt van prototypes of quick-and-dirty demonstratieversies, want daarmee oogst je slechts domme kritiek. PRODOSTA wordt ijverig gebruikt om rookgordijnen op te trekken, het 'echte werk' kun je immers toch niet in zo'n baseline kwijt. Er zijn tegenwoordig allerlei technische mogelijkheden om een systeem flexibel te maken en van vrijheidsgraden te voorzien. De ontwerpers maken daarvan slechts spaarzaam gebruik. Hun ontwerp is immers prima zoals het is, al die flexibiliteit kost performance, vergt ingewikkelde gebruikersdokumentatie en kan in de handen van leken alle maar leiden tot suboptimaal gebruik.

Als het systeem tenslotte vol trots wordt opgeleverd blijken er nauwelijks keuzemogelijkheden voor de organisatie te zijn. Er breekt dus een golf van kritiek los. De ontwerpers zitten nu zelf met de gebakken peren, een vracht werk blijkt voor niets te zijn gedaan. Programmeurs gaan mopperend werken aan de realisatie van een schier eindeloze reeks wijzigingsvoorstellen. 'Als ze dat nu meteen hadden bedacht, dan hoefden we onze fraaie systeemstructuur niet te verpesten.' De schuld ligt uiteraard bij de anderen.

4.4 Mogelijke problemen in de relatie Management-Gebruikers

Bij de start van het projekt is het management slechts geïnteresseerd in de beantwoording van drie simpele vragen: wat kost het, hoeveel levert het op en wanneer is het klaar. Imponderabele doelstellingen zijn bedenkens van fantasierijke hobbyisten, en doen dus niet ter zake. Psychologische overwegingen tellen al helemaal niet mee. Het gevolg is dat de promotors van het nieuwe projekt zich gedwongen voelen een positieve kosten/baten balans in elkaar te draaien zonder daarop PM posten te laten meetellen.

Er is nog één sceptische staffunktionaris die tijd vraagt om even uit te zoeken wat dit nu allemaal gaat betekenen voor het funktioneren van de ploegenwerkers in plant P, om maar niet te spreken van het administratief personeel in afdeling A. Hem wordt te verstaan gegeven dat zulks in een later stadium zeker de nodige aandacht zal krijgen.

Met een zucht van verlichting sluit men de (ten onrechte) als tijdrovend ervaren voorronde af. Het management is het praten over problemen meer dan zat en wendt zich daadkrachtig en oplossingsgericht tot de automatise-afdeling. Fix it, boys! Het projekt komt dus op gang, maar het management is niet geïnteresseerd in het gekompliceerde gebeuren dat door hen-zelf is opgeroepen. Het gaat immers alleen maar om capaciteit erin = resultaten eruit? Er is een schier onuitroeibaar misverstand dat processen kunnen worden beheerst door te draaien aan invoer- en uitvoerkranen, zonder aandacht voor het proces zelf.

Tot overmaat van ramp wordt de bij de toekomstige gebruikers aanwezige kennis en ervaring niet benut, doordat het management toestaat dat de systeemspecificaties buiten hen om tot stand komen. Zij voelen zich dus buiten spel gezet en oordelen schamper over het management: 'het is hùn prestigeprojekt, we zien wel waar het schip strandt.' Hun 'resistance to change' neemt toe, als ze zien hoe het management zich door mooie beloften van de systeem-ontwerpers in de luren laat leggen.

Enkele gewetensvolle gebruikers wijzen hun baas er op dat het nieuwe systeem ook veranderingen in de organisatie met zich mee zal brengen. Het management vindt echter dat we er nu niet óók nog eens een reorganisatie bij kunnen hebben.

In de implementatiefase culmineert de gebruikersweerstand tot een faktor waarmee wel rekening *moet* worden gehouden. Het management doet dat dan ook, maar wel te laat. De implementatie blijkt een tijdrovende zaak te worden, die helaas nauwelijks was gepland waardoor de overeengekomen kosten/baten verhouding grondig uit balans wordt getrokken. Deskundige training is bijv. niet voorzien, dat gebeurt dus à l'improviste met voorspelbare gevolgen voor de gebruikersacceptatie.

Overal in de organisatie ontstaan nu capaciteitsproblemen, het naast elkaar runnen van het oude en het nieuwe systeem leidt tot een bijna ondraaglijke last. Een akelige bijkomstigheid van het gebrek aan planning is ook dat op de ontwerpers in dit stadium geen beroep meer kan worden gedaan; ze zijn immers al weer ingeschakeld bij nieuwe projecten.

Als het systeem een jaar in gebruik is, blijkt dat het management inmiddels zo druk bezig is met weer andere nieuwigheden, dat er van evaluatie of toetsing helaas niets is gekomen.

4.5 Mogelijke problemen in de relatie Management-Ontwerpers

De stafafdeling voor automatische gegevensverwerking bestaat uit informatica-specialisten. Reuze nuttig, maar je moet ze niet van hun werk houden door ze te betrekken in allerlei organisatorische of beleidsproblemen, waarvan ze toch geen verstand hebben.

Door dit uitgangspunt te huldigen, bevestigt het management de ontwerpers in hun vaktechnische professe, en gaat de interessante mogelijkheid verloren dat zij wellicht advies zouden kunnen geven inzake beleidsalternatieven. Dit wordt budgettair onderstreept, door aan de projectleider slechts die armslag te geven die nodig is voor het technische deel van de projektactiviteiten. Het projekt verloopt dus langs een smal, technisch spoor en het management stelt zich daartegen niet kritisch op, integendeel: dit wordt een overzichtelijke, eenvoudig te managen en dus prettige situatie gevonden.

Er zijn weliswaar stuurgroepvergaderingen, maar de onbegrijpelijke taal die daar wordt gehanteerd door systeemontwerpers is niet om aan te horen. Beleidsverwegingen komen dus pas ter sprake als het management weer 'onder ons' is, niet door kennis van zaken gehinderd. Dat geeft de ontwerpers dan weer de kans het projekt volgens eigen voorkeuren te sturen, en het

management onkundig te laten van keuzemogelijkheden in de ontwerpfasen.

Het management is zich vaag bewust dat het systeemontwikkelingswerk eigenlijk zou moeten worden voorgekalkuleerd en getoetst op efficiency en effectiviteit, net als bij een fabriek. Maar wat bij een produktieproces zo eenvoudig is, gaat bij die rare kreatievelingen moeilijk, dus gebeurt het maar niet.

De ruimte die genoemde stafafdeling krijgt is het gemakkelijkst onder controle te houden door van het standpunt uit te gaan dat ze niet meer dan een vast percentage van de omzet mogen kosten. Als er geen nieuwe ontwikkeling aan de gang is hebben ze dus weinig te doen, en als er een groot projekt loopt dan gebeurt dat door ontbrekende capaciteit aan te vullen met inhuurkrachten. Die verdienen zeker 50% meer dan de eigen mensen, zijn dus betere vaklui, en nemen het projekt in feite in handen, zonder verantwoordelijkheid voor de continuïteit.

4.6 Mogelijke problemen in de relatie Management-Hulpbronnen

Voordat het projekt van start gaat zou een inbreng van organisatiekundigen wel nuttig zijn, maar er is over dat nieuwe automatiesysteem al veel te lang gepraat, het is volstrekt duidelijk dat het er moet komen. Om tot die konklusie te komen heeft een beetje directeur echt geen organisatie-advies nodig, laat staan dat er ook nog een tijdrovende gedragswetenschapper aan te pas zou moeten komen.

De enthousiaste projekt-pushers, die intussen al bezig zijn met het doorbladeren van prospektussen van hardware-leveranciers, draaien een kloppend kosten/baten verhaal in elkaar. Helaas wordt vergeten een beroep te doen op professionele voor-kalkulatorische inbreng.

Eén van de belangrijkste hulpbronnen van het management is de macht om het projekt kwa doorlooptijd in te delen. Die macht wordt in het algemeen slecht gebruikt. Een nerveus verlangen om de oplossing in technische zin snel naderbij te brengen zorgt ervoor dat de PRODOSTA fasen 1 en 2 worden afgeraffeld, met kwalijke gevolgen voor kwaliteit en akseptatie van de gemaakte keuze. Dit schijnt typerend te zijn voor het Westerse denken. Japanners zijn verbijsterd als ze worden gekonfronteerd met de onverantwoord korte tijd die wij uittrekken voor oordeels- en besluitvorming.

Het management rekruteert de projektleider bij voorkeur uit de kring van informatica-specialisten. Bij alle onzekerheid op dit moeilijke vakgebied is het immers een veilig idee als het projekt door een vakman wordt gerund. Of hij veel ervaring heeft in werkkoördinatie en planning is allerminst zeker. Of hij ook nog enig benul heeft van onderhandelen, procesbegeleiding en kennisoverdracht is een vraag die niemand stelt. De enigszins onzekere projektleider doet alles graag volgens het boekje, maar er is méér nodig dan een formele methodiek als PRODOSTA, om greep te krijgen op kommunikatie en samenwerking van mensen.

Van de vele mogelijkheden tot projektstructurering waarover het management beschikt wordt er slechts één gebruikt: er komt een stuurgroep, die de projektleider in het goede spoor moet houden. Helaas heeft

het management geen tijd om daar zelf in te gaan zitten. Het wordt dus een kibbelende commissie van een aantal bij het project betrokken afdelingschefs, die zich zien als partijen in een onderhandelings-situatie, met de specialist/projectleider als neutrale deskundige. Hij neemt dus de feitelijke leiding en stuurt de stuurgroep, in plaats van andersom.

Als het management zo nu en dan in een opwelling van bezorgdheid vraagt of alles wel volgens PRODOSTA verloopt, dan wordt geruststellend gewaardeerd met dikke baseline-dokumenten. De projectleider schrijft een management-summary, met een knipoog naar de stuurgroepleden die allang blij zijn dat er iemand is die het management bekwaam zoet houdt.

5 VALKUILEN OP WEG LANGS DE 'PROJECT LIFE CYCLE'

Wij hadden, zoals in het vorige hoofdstuk reeds is opgemerkt, een lijst opgesteld van circa 80 knelpunten of belemmeringen, die bij een automatieproject een rol kunnen spelen.

In hoofdstuk 4 zijn deze knelpunten verwerkt in een zestal relatie-beschrijvingen.

Dezelfde knelpuntenlijst kon ook worden gehanteerd om per fase in de PRODOSTA life cycle, dus langs de tijdas, waarschuwend signalen te plaatsen. Dat is gebeurd in dit hoofdstuk.

5.1 PRODOSTA fase 1: Vooronderzoek

WAT IS HET BEDRIJFSPROBLEEM?

Bij de formulering van het 'business problem' wordt vaak vergeten een duidelijke relatie te leggen naar de doelstellingen van de organisatie. Men projekteert de vraag naar een automatisatiesysteem op de situatie ad hoc, zonder rekening te houden met het meerjarenplan en de ontwikkelingen op langere termijn. Daarbij verkijkt men zich vaak op de tijdsduur die zal verlopen tussen projectstart en oplevering van het informatiesysteem.

Het is van belang om het bedrijfsprobleem 'clean' te analyseren, en daarbij aan overwegingen van technische, economische, sociale en organisatorische aard elk het volle pond te geven.

Uit ongeduld gaat men vaak oplossingsgericht denken, hoewel dat in fase 1 taboe zou moeten zijn. Oplossingsgericht denken leidt onvermijdelijk tot overaccentuering van het informatie-technische aspect. Men verkrijgt dan een informatica-probleemstelling die niet wortelt in meer algemene overwegingen van beleid.

Het is de kunst het informatieprobleem als deelprobleem te definiëren, in het kader van het 'business problem', om het pas daarna te isoleren als probleemstelling voor fase 2.

5.2 PRODOSTA Fase 2: Toepassingsanalyse

WELKE IS DE OPTIMALE OPLOSSING VOOR HET INFORMATIE-PROBLEEM?

Dit is waarschijnlijk de moeilijkste en belangrijkste fase in de life cycle. Er wordt een beroep gedaan op

kreativiteit, analytisch vermogen en de vaardigheid tot konstruktieve planning.

Alternatieve oplossingen voor het informatieprobleem moeten worden bedacht (kreatief denken). Uit die alternatieven moet de beste oplossing naar voren komen (analyse). De weg naar de gekozen oplossing moet worden ingeslagen door het opbouwen van een planning en het bepalen van de projectstructuur.

Veelal neemt men genoeg met één of slechts enkele suggesties, zonder tijd en energie vrij te maken voor het genereren van meer alternatieven. Denken volgens vaste patronen (zó pak je dat nu eenmaal altijd aan) blokkeren de creativiteit.

Als het aankomt op weging van de alternatieven dan moet niet worden vergeten om naast techniek en economie ook te denken aan sociale en organisatorische consequenties. Actieve betrokkenheid van het management en van de gebruikersorganisatie is hierbij van essentieel belang. Als dat wordt vergeten dan is men afhankelijk van de oordeelsvorming in de kring van systeemontwerpers. Dat hoeft niet tot een slecht systeem te leiden, maar de kans op een optimale keus is klein, en bovendien wordt dan de basis gelegd voor onvolledige implementatievoorbereiding.

Tenslotte wordt vrijwel altijd verzuimd de projectstructuur zorgvuldig op te bouwen. Iemand wordt haast-je-rep-je opgezadeld met het projectleiderschap, op basis van een gegokt budget en met de opdracht om zelf maar te zien waar de benodigde werkkraft en know-how vandaan moeten komen.

Een beknopte checklist voor projectstructurering:

Stuurgroep, projectteam, werkgroepen: taken en verantwoordelijkheden.

Keuze van de projectleider bij voorkeur uit de kring van de toekomstige systeemgebruikers, met als primair vereiste deskundigheid projectmanagement (informatica is ook nuttig; maar komt niet op de eerste plaats).

Fasering en planning kwa:

- inhoudelijke volgorde
- overleg en besluitvorming met de gebruikers
- benodigde deskundigheden
- benodigde capaciteit in geld en doorlooptijd.

Pas als deze infrastructuur voor de verdere projectgang is voltooid, kan men met goed vertrouwen de gekozen oplossing in fase 3 gaan specificeren.

5.3 PRODOSTA Fase 3: Systeemspecificatie

HOE ZULLEN DE TOEKOMSTIGE GEBRUIKERS HET SYSTEEM ZIEN?

In deze fase nemen de ontwerpers vaak de sturing over. Het begint 'technisch' te worden. Het grote gevaar in fase 3 is het afhaken van management en gebruikers. Zij zijn juist in fase 3 nodig om het besef

levend te houden dat het niet primair gaat om een technische ontwikkeling, maar om een organisatieverandering.

Er moet aandacht zijn voor informatie van en overleg met de toekomstige gebruikers.

Specificaties van het 'User/System Interface' moeten worden getoetst op bruikbaarheid, zonodig met behulp van pilot implementaties voor moeilijk begrijpelijke deelsystemen. In de System Requirements Specification moeten niet alleen de eisen aan datastructuur en programma's uit de verf komen, maar ook de toekomstige werkwijze en spelregels.

Een extra moeilijkheid ontstaat waar (zoals steeds vaker gebeurt) een pakket-implementatie de hoofdmoot van het project is. Het komt er dan op aan om de mogelijkheden tot parametrisering van het pakket en aanvullende programmering zorgvuldig te benutten, en voor het overige precies vast te stellen waar de organisatie zich procedureel zal moeten aanpassen.

De gevaarlijkste valkuil in fase 3 is, dat men geneigd is het niet al te nauw te nemen met de gebruikersacceptatie. Een handtekening van het management is waardeloos tenzij de gebruikers ten volle overzien wat hun boven het hoofd hangt en hun commentaar is verwerkt. Als ze niet willen of kunnen lezen, dan moeten er maar presentaties, demonstraties en rollenspellen aan te pas komen. Ze moeten het snappen, anders is het zinloos om fase 4 in te gaan.

5.4 PRODOSTA Fase 4: Systeem definiëring

HOE ZAL HET SYSTEEM TECHNISCH IN ELKAAR GAAN ZITTEN?

Als de eerste drie fasen zorgvuldig zijn doorlopen, dan kan er in fase 4 weinig misgaan. Het gaat nu inderdaad om het vakmanschap van de ontwerpers. Een tijdelijke vermindering van de bemoeienis van gebruikers en management hoeft geen ramp te zijn.

Een mogelijk knelpunt in deze fase is de afwezigheid van voldoende informatie-technische kennis. Dat leidt dan of tot een slecht ontwerp, of tot het inschakelen van hoogwaardige krachten op tijdelijke basis. Als minste van twee kwaden is het laatste te prefereren, maar dat roept wel het gevaar op dat het systeem straks een onbegrepen verdeling wordt, verlaten door de inmiddels elders te werk gestelde ontwerpers.

5.5 PRODOSTA Fase 5: Systeembouw

KONSTRUKTIE EN IMPLEMENTATIE.

Dit is de fase waarin de planning uitloopt, de kosten de pan uitrijzen en de misverstanden kulmineren tot conflicten. In de voorafgaande fasen is (in het ergste geval) verzuimd om de implementatie te plannen, de gebruikers voor te bereiden en de organisatorische consequenties onder ogen te zien.

Dit leidt dan tot 'resistance to change' en overdoen van ontwikkelwerk. Vaak komt men in een eindeloos iteratieproces terecht, waarbij de systeembouwers veranderingen blijven aanbrengen en de gebruikers het systeem blijven afkeuren. Dit moeizame proces wordt bovendien nogal eens slecht begeleid.

Om er iets van te maken moet worden gedacht aan deskundigheid op de volgende gebieden:

- organisatie verandering
- sociale begeleiding
- kennisoverdracht.

5.6 PRODOSTA Fase 6: Operationeel systeem

ZIJN DE DOELSTELLINGEN BEREIKT?

Het grote probleem met deze fase is, dat veelal totaal wordt vergeten om deze vraag aan de orde te stellen. Fase 6 is de evaluatie-fase, en evaluatie is objectieve toetsing aan eerder geformuleerde doelstellingen. Dat betekent onderzoek naar gebruikerservaring en formuleren van sterke en zwakke punten in het geïmplementeerde systeem.

Niet alleen de eerder genoemde doelstellingen zijn voor deze toetsing van belang. Het komt vaak voor dat er onuitgesproken doelstellingen blijken te zijn nagestreefd door sommige belanghebbenden. Dat soort 'illegale' doelstellingen komen pas na de implementatie aan het licht. Zelfs is het mogelijk dat er neveneffecten ontstaan die door niemand zijn voorzien, een soort creaties vanuit de konfrontatie tussen de organisatie en het in gebruik gestelde systeem. 'Toen het er eenmaal was hebben we ontdekt dat we het systeem daar óók voor konden toepassen.' Onbedoelde functies hoeven niet a priori ongewenst te zijn. Het is wel zaak ze te onderkennen en in de evaluatie mee te nemen.

In de operationele fase moet er aandacht zijn voor het beheer, en dus voor de effectiviteit van het hanteren van het systeem. Het gaat daarbij om

- functioneel beheer (doet het systeem wat we willen?)
- technisch beheer (maintenance op programmatuur en files)
- operationeel beheer (werkt men volgens afspraak?)

Voor elk van deze beheersvormen moet mankracht aanwezig zijn van voldoende bekwaamheid. Het gevaar is dat men het informatiesysteem ziet als een inmiddels verworven deel van de infrastructuur van het bedrijf, waaraan verder geen geld of aandacht hoeft te worden besteed. Niets is minder waar. Men mag blij zijn als de jaarkosten in de operationele fase niet meer dan een derde bedragen van wat er tijdens de projectfasen 1 t/m 5 op jaarbasis werd besteed.

6 WEGEN TOT VERBETERING

De voorafgaande hoofdstukken 4 en 5 hebben een signalerend en probleembeschrijvend karakter. De problematiek van ineffektieve automatieprojecten werd onder woorden gebracht in termen van verstoorde relaties tussen de participanten in het project (hoofdstuk 4) en door het zetten van een reeks waarschuwborden langs de 'Project Life Cycle' route (hoofdstuk 5).

We hebben ons niet daartoe willen beperken, en komen daarom in dit hoofdstuk tot een aantal aanbevelingen om in de toekomst automatieprojecten beter te doen verlopen.

In de meest beknopte vorm kunnen onze konklusies worden samengevat in drie basisprincipes.

- 1 Een automatieproject moet worden gezien als een organisatieveranderingsproces.
- 2 Participatie van de gebruikers van het te implementeren informatie-systeem is essentieel voor het slagen van het project.
- 3 De beschikbare instrumenten (PRODOSTA, project planning e.d.) en de beschikbare deskundigheden (informatie-technisch, organisatorisch, sociaal e.a.) moeten optimaal worden benut.

Deze drie basisprincipes vinden een nadere uitwerking in de nu volgende aanbevelingen. Deze zijn bedoeld als punten waarop betrokkenen bij een automatieproject zouden moeten letten.

6.1 Aanbevelingen aan het management

- 1 Beschouw automatisering als organisatieverandering en beoordeel het te bouwen informatiesysteem als iets wat grote invloed zal hebben op het functioneren van uw organisatie als totaal.
Wees u ervan bewust dat niet alleen kwantificeerbare maar ook imponderabele opbrengsten en offers voor uw besluitvorming van belang zijn.
Betrek niet alleen technische en economische, maar ook sociaal/organisatorische overwegingen bij uw oordeelsvorming.
- 2 Bevorder vanaf het eerste begin de betrokkenheid van de eindgebruikers en maak gebruik van hun kennis en ervaring; beschouw de acceptatie van de eindgebruikers als een onmisbare sleutel tot een goed resultaat.
Verwacht niet dat het automatieproject een gedetermineerd proces zal zijn nadat u het op de rit hebt gezet; uw sturing en actieve betrokkenheid blijven nodig gedurende de hele project life cycle.
- 3 Bevorder korrekt gebruik van PRODOSTA, onderken daarbij dat met name de fasen 1, 2 en 5 voor een goed resultaat - gezien vanuit uw organisatie - het belangrijkste zijn.
Schakel naast de informatica deskundigheid ook andere voor een goede projectvoering benodigde deskundigheden in, zoals voorkalkulatie, sociale begeleiding, bedrijfskundig advies, project management, opleiding . . . ; doe dit vooral tijdig, en maak vanaf de start van het project duidelijk wat ieders taak en verantwoordelijkheid is.
Benoem de projectleider bij voorkeur uit de gebruikerskring, eis 'Project Management' als zijn primaire deskundigheid en beschouw informatica-kennis als een voor de projectleider nuttige zaak van lagere prioriteit.

6.2 Aanbevelingen aan de gebruikers

- 1 Beschouw het automatieproject als een veranderingsproces, met effect op de voorwaarden

waaronder u in de toekomst uw werk zult doen. Geef u rekenschap van de onzekerheden die deze verandering met zich mee brengt voor het functioneren van u en uw organisatie.

- Verlang actieve betrokkenheid van uw management.
- 2 Onderken vanaf het begin het belang van uw actieve participatie; u hebt recht op informatie, daartegenover staat dat u er extra energie moet instoppen in uw eigen belang, gezien de mogelijke effecten van het project.
Laat niet alles over aan systeemontwerpers en management, maar reduceer van uw kant de onzekerheid door uw systeemeisen te specificeren. Waar onzekerheden onvermijdelijk zijn moet u op voldoende systeemflexibiliteit aandringen opdat in een later stadium aan uw nog niet specificerbare behoeften tegemoet gekomen kan worden.
Bring aan op fasering van het project en een daaraan gekoppelde besluitvorming; gebruik uw mogelijkheden tot werkoverleg voor informatie-uitwisseling en het 'inregelen' van uw vertegenwoordigers in het project.
 - 3 Zie PRODOSTA niet als een exclusief tool van de ontwerpers, maar als een hulpmiddel dat ook u ten dienste staat.
Maak uzelf een voorstelling van wat van u verlangd zal worden als het systeem klaar is, dat voor u een stuk gereedschap moet zijn.
- #### 6.3 Aanbevelingen aan de ontwerpers
- 1 Realiseer u dat u met uw werk zeer sterk kan ingrijpen - positief en negatief - in het klimaat en de effectiviteit van de organisatie waarvoor u werkt.
Zie het systeem dat u bouwt niet als uw exclusief geestelijk eigendom, maar als een stuk gereedschap voor de gebruikersorganisatie.
Erken het management van de gebruikersorganisatie als de leidinggevende instantie, ook wat uw projektactiviteiten betreft.
 - 2 Luister goed naar de gebruikers, ook als ze geen of weinig verstand van informatica hebben.
Investeert in systeemflexibiliteit; het is dan later op grond van de eerste ervaringen met het systeem nog mogelijk fouten of vervelende gevolgen voor de gebruikers te herstellen.
Wees vooral kritisch op de kwaliteit van probleemanalyse en applicatievoorstel en vergewis u ervan of dit voldoende breed gedragen wordt in de organisatie.
 - 3 Accepteer de invloed van deskundigheid van andere staforganen of adviseurs, zie hen niet als concurrenten op uw terrein maar als instanties die kunnen bijdragen tot grotere effectiviteit van uw werk.
Gebruik PRODOSTA korrekt, dat wil zeggen als een middel tot informatie-uitwisseling en projektbeheersing voor uzelf, voor het management en voor de gebruikers.
Zorg dat de organisatie u stuurt en niet andersom.

27 Een socio-technische aanpak: automatisering voor de rechter hersenhelft

door A. Kranendonk R.A.

Dit artikel schetst de problemen waarin de automatisering zich thans bevindt. Om deze fundamenteel op te lossen, is een aanpak van automatiseringsvraagstukken gewenst, waarin alle aspecten van de problematiek - zowel de 'sociale' als de 'technische' - op een gelijkwaardige wijze in de besluitvorming worden betrokken. Een aanpak waarbij kennis van de onderdelen vergezeld gaat van gevoel voor het geheel.

Niet in alle organisaties is een dergelijke socio-technische aanpak van automatisering (STAA) mogelijk. Hij heeft alleen kans van slagen, als de mensen die de organisatie vormen dat ook willen.

Dat betekent, dat men met betrekking tot het te ontwikkelen systeem de eisen wil en kan laten formuleren door alle betrokkenen. In het ontwikkelingsproces zal men bereid moeten zijn te leren van en met elkaar.

1 INLEIDING

De huidige stand van zaken in de automatisering geeft reden tot bezorgdheid: enerzijds lopen projecten qua kosten en/of tijd enorm uit, anderzijds zijn management en gebruikers niet in de mate die mocht worden verwacht, betrokken bij de ontwikkeling van informatiesystemen en gemotiveerd voor het gebruik daarvan.

Complexe systemen komen moeilijk van de grond en sluiten slecht aan op de veranderende informatiebehoeften van gebruikers. Bovendien vreest en verwacht men door de automatisering ongewenste gevolgen voor de werkomstandigheden. Kortom: een voor automatiseringsspecialisten, ontwikkelaars, managers en gebruikers onbevredigende situatie.

De automatiseringsspecialisten trachten aan deze situatie een einde te maken door steeds verfijndere methoden voor systeemontwikkeling te propageren, maar ook door te streven naar een meer organisatiegerichte aanpak van automatisering: 'informatiebehoeften van de gebruiker' en 'gebruikersparticipatie' zijn daarbij graag gehanteerde begrippen.

Wat hiervan echter in de praktijk terecht komt, valt nog te bezien. Zeker, in projektorganisaties wordt veel gedaan om de managers en de gebruikers bij de ontwikkeling 'te betrekken' en hun informatiebehoeften op tafel te krijgen. Maar wordt een daadwerkelijk participatieve aanpak bereikt? Ondanks alle goede bedoelingen is het meedenken en meebeslissen door managers en gebruikers vaak nog een ideaal beeld. Er zijn kennelijk ondanks het *willen*, problemen met het *kunnen*.

In dit artikel worden enkele kaders aangegeven die het *kunnen* mogelijk maken. Na een verkenning van het probleemveld in paragraaf 2, zal in een tweetal hoofdlijnen een fundamenteel andere aanpak van automatiseringsvraagstukken worden gegeven: het be-

Motto's:

These concepts are useful, but only in a limited context. That context is when physicists, for convenience, pretend, as well all do, that dancers can exist apart from a dance.

Gary Zukav

Uit: The Dancing Wu Li Masters - An Overview of the New Physics

*Labour is blossoming or dancing where
The body is not bruised to pleasure soul,
(...)*

*O body swayed to music, O brightening glance,
How can we know the dancer from the dance?
William Butler Yeats*

Uit: Among School Children

schouwen van automatisering als socio-technische organisatie-ontwikkeling.

In paragraaf 3 wordt beschreven *wat* met een Socio-Technische Aanpak Automatisering* wordt ontwikkeld: het *produkt*.

Het *hoe* van deze aanpak wordt uiteengezet in paragraaf 4: het *proces*.

Tenslotte wordt in paragraaf 5 een brug geslagen tussen de in dit artikel beschreven ideale situatie en de *praktijk* van alle dag.

De titel van dit artikel - een variatie op de naam van

* STAA kreeg gestalte door een langdurig contact van Klynveld Kraayenhof & Co., Organisatiegroep (KKC-OG) met het Institute for Social and Organizational Development (IOD) te Leuven, door praktijkervaringen en door discussies met vakgenoten binnen en buiten de Organisatiegroep. Dit artikel kwam voor een deel tot stand door het selecteren en redigeren van het vele materiaal dat de IOD-KKC-groep in de loop der jaren over STAA heeft geproduceerd en mondjesmaat gepubliceerd[7][26].

een cursusboek voor creatief tekenen [8] - legt de relatie tussen automatisering en de manier waarop onze hersenhelften werken.

'Men vermoedt dat de hersenhelften verschillende strategieën of tactieken hanteren: de linker maakt gebruik van een analytische, verbale, sequentiële, lineaire benadering, terwijl de rechter meer 'synthetisch', 'holistisch' en niet-lineair zou werken. Bij het waarnemen zou zij bijvoorbeeld niet zozeer details als wel gehele 'Gestalten' percipiëren[19].

Een deel van de huidige problematiek in automatiseringsland kan wellicht worden verklaard uit het verleden. In de begindagen van de computer lag alle aandacht op het ontwikkelen van foutloze programma's, het zorgen voor 'schone' invoer en het operationeel houden van het systeem. Een taak waarbij de linker hersenhelft volop aan bod kwam. Van lieverlede wordt mét de toenemende technische mogelijkheden de invloed van de computer op de organisatie groter. De eenduidige, alle kondities omvattende modellen van de automatiseringsspecialisten blijken daarbij slecht vertaalbaar te zijn naar het organisationele niveau: mensen in organisaties handelen lang niet altijd volgens beslissingstabellen!

Naast de analytische, rationele, stap-voor-stap werkende linkerkant komt er dan ook steeds meer behoefte aan het creatieve, gevoelsmatige zien-in-gehelen van de rechterkant.

Daarom is dit artikel bestemd voor de rechter hersenhelft van alle (gedeeltelijk overlappende) groepen mensen in de organisatie die bij automatisering zijn betrokken (zie figuur 1):

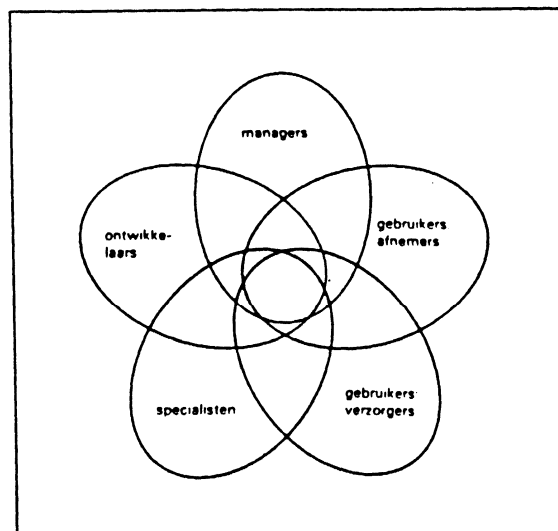
- de automatiseringsspecialisten (die alles weten van apparatuur, programmatuur, methoden en technieken);
- de systeemontwikkelaars (die informatiesystemen bouwen, die apparatuur/programmatuur kiezen/maken, die procedures bedenken, die veranderingen in de organisatie bewerkstelligen);
- de managers (die de eindverantwoordelijkheid hebben voor het reilen en zeilen van de organisatie: beleid en dagelijkse gang van zaken);
- de gebruikers:
 - de informatie-afnemers (die de informatie bij hun werk nodig hebben) en
 - de informatieverzorgers (die het systeem onderhouden en met gegevens voeden alsmede in de informatiebehoeften van de afnemers voorzien).

Kortom, een artikel voor de mens in de organisatie. Daarbij wordt, voor het gemak, soms gedaan alsof 'mensen' en 'organisatie' verschillende entiteiten zijn; wie de motto's aanvoelt, weet wel beter.

2 PROBLEEMVELD

Het probleemveld 'automatisering' wordt in het volgende via drie invalshoeken bekeken:

- waar het allemaal om is begonnen: het informatiesysteem,
- de manier waarop dat systeem tot stand komt: het ontwikkelingsproces en



Figuur 1: De Betrokkenen

- de organisatie waarvoor dat systeem wordt ontwikkeld.

2.1 Het informatiesysteem

Een (gedeeltelijk geautomatiseerd) systeem van gegevensverwerking en informatievoorziening is geen doel op zichzelf: een informatiesysteem heeft een ondersteunende functie in de strategische, tactische en operationele taken van een organisatie.

De vele ontwikkelingsmethoden in de automatisering laten niet alleen een brede scala zien van manieren om geautomatiseerde informatiesystemen te ontwikkelen, maar geven daarbij óók verschillende modellen van wat met methode 'x' of 'y' moet worden ontwikkeld. Bovendien hebben de bij het te bouwen informatiesysteem betrokken partijen (specialisten, ontwikkelaars, managers en gebruikers) 'een eigen perceptie ofwel model daarvan'[1a]. De te gebruiken methode moet daarom 'de mogelijkheid bieden deze verschillen te overbruggen, dus te komen tot een gemeenschappelijk aanvaarde conceptie'[1b]. Wat een 'goed' of 'waar' model is, kan dus niet rationeel worden bewezen: een model is waar, zolang de resultaten van het werken daarmee voor alle betrokken partijen bevredigend zijn.

De beelden van hetgeen in de diverse methoden wordt ontwikkeld en wat dat betekent voor een organisatie, verschillen nogal qua uiterlijk, maar bewegen zich qua achterliggende filosofie tussen de ideologische polen: 'mensen zijn lastig' en 'mensen zijn noodzakelijk'.

De eerste opvatting probeert de mens (een subjectief en onvoorspelbaar element) te weren uit het denken over organisaties en informatiesystemen. Het heet dan, dat 'de relevante structuur van een organisatie vaak wordt versluierd door subjectief menselijk denken over organisaties[30]. De gegevensstructuur zou onafhankelijk van het individuele gebruik daarvan moeten worden bepaald (omdat dit steeds kan verschillen), maar 'uitgaande van de afzonderlijke gegevens en hun afhankelijkheden'[20]. Een 'gebruiker'

wordt soms omschreven als 'iemand of iets, die berichten zendt aan het informatiesysteem, of hieruit berichten ontvangt'[13]. Het gevaar is dus niet denkbeeldig, dat bij het ontwikkelen van informatiesystemen, de mens als een soort machine wordt gezien.

Toch zijn er in de literatuur ook andere opvattingen: 'Er is bijvoorbeeld geen objectieve maatstaf voor de doeltreffendheid van een informatiesysteem'[4]; een veranderingsanalyse is de eerste stap in het ontwikkelingstrajekt, dat zich over een breder terrein uitstrekt dan de bouw van een informatiesysteem[22]; er is grote behoefte 'aan een meer geïntegreerde en multidisciplinaire aanpak van organisatie- en informatieproblemen'[30]; inzicht en motivatie van de mensen staan centraal, methoden en technieken zijn slechts hulpmiddelen[27]; een organisatie is een informatiesysteem, dat wil zeggen, heeft naast andere functies ook de functie van informatiesysteem: 'het informatiesysteem is een aspectstelsel van een organisatie'[29].

Dit betekent, dat informatiesystemen moeten worden ontwikkeld als organisaties. Echter: 'Kijken naar organisaties kan op verschillende manieren. Is het een gebouw, waar 's morgens grote aantallen mensen naar binnen gaan om het 's avonds weer te verlaten? Is het een grote 'familie' van kollega's, chefs, ondergeschikten, kortom, medewerkers? Een samenstelsel van mensen en middelen die met elkaar iets voortbrengen? Een instituut om geld te verdienen? Een verzameling arbeidsplaatsen?'[6a]

Er zijn verschillende beschouwingswijzen van een organisatie mogelijk en daarmee dus ook van een informatiesysteem. Het model van een organisatie dat men voor ogen heeft, beïnvloedt de mate waarin het ontwikkelde systeem in die organisatie 'past'. Een door de managers en de gebruikers ongewild en onbegrepen systeem wordt gebrekkig gehanteerd (of erger, wordt gesaboteerd): het blijft een boeteersel waaraan de levensadem ontbreekt. Informatiesystemen moeten dus breed worden gespecificeerd. Anders zal er een onvoldoende afstemming blijven tussen de wensen van de organisatie en de realiteit van het informatiesysteem, tussen behoeften en mogelijkheden.

2.2 Het ontwikkelingsproces

De diverse ontwikkelingsmethoden schenken naast de zorg voor de inhoud van het te ontwikkelen systeem ruime aandacht aan het ontwikkelingsproces. Er zijn vele fasen en stappen met elk hun eigen meetpunten, bijvoorbeeld 'de mijlpalen waar er een tussenkomst vanwege de hogere directie noodzakelijk is'[12]. Er wordt soms een zwaar accent gelegd 'op adequate participatie van (top)management, als instantie die geroepen is om een uitspraak te doen over(...) doelstellingen, over de aan de systemen te stellen eisen en hun relatieve prioriteiten'[23].

De samenwerkingsproblemen in het ontwikkelingsproces krijgen minder aandacht. Zeker, er wordt wel onderkend, dat naarmate het te ontwikkelen systeem een bredere strekking heeft in de organisatie, bij het ontwikkelingsproces meer deskundigheden en verantwoordelijkheden (disciplines) nodig zijn om het systeem te realiseren. Echter, de

daaruit voortvloeiende samenwerkingsproblemen worden in de diverse ontwikkelingsmethoden meestal niet realistisch opgelost. Dat elke discipline verschillende doelstellingen nastreeft, wordt (gelukkig!) soms genoemd; de hantering van deze strijdige belangen via bijvoorbeeld een puntentoekenning[2] suggereert echter méér harmonie tussen participanten dan in de praktijk voorkomt. Wie immers bepaalt wat het belangrijkste is? Het op elkaar afstemmen van automatiserings-technische, sociale, organisatorische en bedrijfseconomische aspecten vraagt qua samenwerkingsvaardigheden méér dan de machtswoorden: 'Daar zij een projektorganisatie!'

De deelnemers aan het ontwikkelingsproces moeten leren samenwerken. Niet alleen in leersituaties, maar ook bij 'het 'Eeuwig Leren' dat praktijk wordt genoemd'[5].

2.3 De situatie in een organisatie

Breed opgezette, geautomatiseerde informatiesystemen omvatten een massaliteit van gegevens en processen, die tijdens het ontwikkelingsproces tot in detail worden gedefinieerd. Ontwikkelingsmethoden helpen daarbij, voor zover deze als activiteit een gestructureerde informatie-/gegevensanalyse bevatten. De voorbeelden die hierbij worden gebruikt, betreffen meestal een voor allen bekend operationeel proces (zoals orderafhandeling en facturering) en de directe besturing daarvan[3]. (Wat dergelijke methoden kunnen betekenen voor de meer tactische en strategisch getinte informatiebehoeften, is nog de vraag.) Het nauwkeurig voorschrijven en uitwerken van allerlei zaken heeft - vaak onbedoeld - een centraliserend effect. Grotere uniformiteit, gewenst vanuit allerlei afstemmingsproblemen in de praktijk en een sterkere, centrale beheersing in ontwikkelingsmethoden staan echter lijnrecht tegenover de huidige tendens naar decentralisatie in organisaties en de verschuiving van een alomvattende centrale leiding naar een grotere autonomie op lagere niveau's. De door sommigen gekonstateerde toenemende kans 'op een konfliktsituatie tussen de centralisatie binnen organisaties en de differentiatie in de omgeving'[15] heeft echter niet op te treden. Immers, hoe complexer en onzekerder de omgeving is, des te meer een organisatie is gebaat bij relatief autonome onderdelen en dus ook informatiesubsystemen. Hierbij past ook het streven van mensen naar meer invloed op hun eigen werk en taakinhoud. De rol van de automatiseringsdeskundige (specialist) verschuift daarbij naar die van dienstverlener (ontwikkelaar). Computers worden dan gezien als hulpmiddelen in een voor en door gebruikers gedefinieerd informatiesysteem.

De afstand tussen het centraal opgezette, alles omvattende, starre systeem en de flexibele organisatie wordt dan verkleind. Een voorwaarde is dan wel, dat er duidelijkheid is over de organisatorische doelen (zoals centraal/decentraal) die de verschillende (groepen van) betrokkenen door middel van automatisering willen realiseren.

2.4 Drie opgaven en een andere aanpak

Ontwikkelingsmethoden staan als we de voorgaande

punten samenvatten, in de huidige automatisering voor drie opgaven:

- een specificatie vanuit de technische én sociale dimensie van het informatiesysteem,
- een oplossing van de samenwerkingsproblemen bij de ontwikkeling en
- een voldoende besef van de situatie in een organisatie: de invloed van de aktualiteit.

Specificatie vanuit de technische en sociale dimensie
De in de praktijk aan informatiesystemen gestelde eisen zijn té vaak alleen gericht op technische, inhoudelijke zaken, zoals: gegevens, te verstrekken informatie, frekwenties en dergelijke. Dat het informatiesysteem de toekomstige taakuitoefening van vele medewerkers van een organisatie bepaalt, wordt door ontwerpers (en gebruikers!) vaak verwaarloosd. Mensen worden dan gezien als rationeel werkende verlengstukken van de techniek: er wordt gesproken van 'ontwerpbeslissingen aangaande de wijze waarop en de mate waarin de menselijke faktor (ook wel 'aktor' genoemd) wordt ingezet als realisatie van een functioneel ontworpen systeem'[21]. Mensen, zo lijkt het wel, moeten allen dezelfde betekenis aan gegevens toekennen en met dezelfde informatie tot (dezelfde?) besluiten komen. Een inkompleet beeld, wat tot onbevredigende resultaten leidt. Mensen zijn méér dan even zovele paren handen. Het ontwikkelen en invoeren van informatiesystemen resulteert in het ontwikkelen van organisaties.

Samenwerkingsproblematiek

Ontwikkelingsmethoden laten veelal in het midden, hoe zwaar de invloed vanuit de verschillende disciplines moet/mag zijn. Het omgaan met de soms tegenstrijdige waarderingscriteria wordt opgelost via ogenschijnlijk objectieve technieken (wegingscriteria, bevoegdheidsmatrices). In de praktijk staat echter tegenover de deskundigheidsmacht van de automatiseringsspecialisten en systeemontwikkelaars de sanktiemacht (passieve rol in het ontwikkelingsproces, mis-gebruik, sabotage) van managers en gebruikers[26].

Een weekje cursus (bij het begin van een groot project) over 'het samenwerken in en met groepen' is niet genoeg om een en ander op te lossen. En misschien zit het probleem wel in de 'samenwerkingsidee'. Immers, waar doelstellingen en belangen strijdig zijn, wordt (openlijk of verholen) onderhandeld over het verloop van het ontwikkelingsproces en de uitkomst daarvan: de vorm en de inhoud van het informatiesysteem.

Invloed van de actuele situatie

De 'rijdende trein' wordt bij de aanvang van een project gebagatelliseerd. Toch kunnen bestaande plannen en visie op automatisering de geloofwaardigheid verkleinen van de bedoelingen die men heeft met participatie. Het automatiseringsverleden van een organisatie kan een niet te onderschatten belemmering zijn bij het beginnen aan iets nieuws. In de loop der tijd zijn vooropgestelde, onuitgesproken meningen (percepties) gegroeid. Als deze onderling afwijkende percepties niet bekend zijn, leiden zij tot onduidelijkheden en dus tot een onbevredigend automatiseringsproces.

Bijvoorbeeld:

- de gebruiker denkt, dat de automatiseringsdeskundige (specialist/ontwikkelaar) alles wil automatiseren om daarmee de eigen positie in de organisatie te versterken;
 - de automatiseringsdeskundige denkt, dat de manager zoveel mogelijk door middel van gecentraliseerde systemen wil automatiseren;
 - de gebruiker denkt, dat de manager zoveel mogelijk arbeid van de mens naar de computer wil brengen;
 - de manager denkt, dat de automatiseringsdeskundige zal zorg dragen voor een eenvoudig, up-to-date, het hele bedrijf omvattend, management-informatiesysteem dat bovendien niet te duur is en efficiënt werkt;
- enzovoort.

Een andere aanpak

In het hiervoor geschetste probleemveld gaat het om fenomenen als 'doelen', 'macht', 'waardering', 'percepties' en dergelijke. Deze kunnen niet vanachter de schrijftafel worden ingeschat. Om de problemen adequaat aan te pakken is een voortdurende waarneming nodig (van normen, waarden, percepties, attitudes, enzovoort) in de organisatie waar automatisering plaatsvindt. Dit kan worden gerealiseerd door middel van *wezenlijke participatie van alle betrokkenen* bij automatisering. Omdat in deze aanpak wordt gestreefd naar een evenwicht tussen de 'sociale' en de 'technische' criteria, een in hun onderlinge samenhang bezien van 'technische instrumentatie en arbeidsverdeling'[24], draagt hij de naam: Socio-Technische Aanpak van Automatiseringsvraagstukken (STAA). Deze aanpak, STAA, is een raamwerk. Het kan worden gekombineerd met elke ontwikkelingsmethode die qua filosofie of ideologie de mens ook als 'geheel' kan beschouwen. (In die zin zouden de letters 'STAA' óók Socio-Technische Aanvulling op Automatiseringsmethoden kunnen betekenen.)

Dat het een raamwerk is, houdt óók in, dat het is beschreven als een ideaal beeld. Per organisatie zal, inspelend op de praktijksituatie, een aangepaste toepassing moeten worden opgezet.

STAA wordt hierna behandeld langs twee hoofdlijnen:

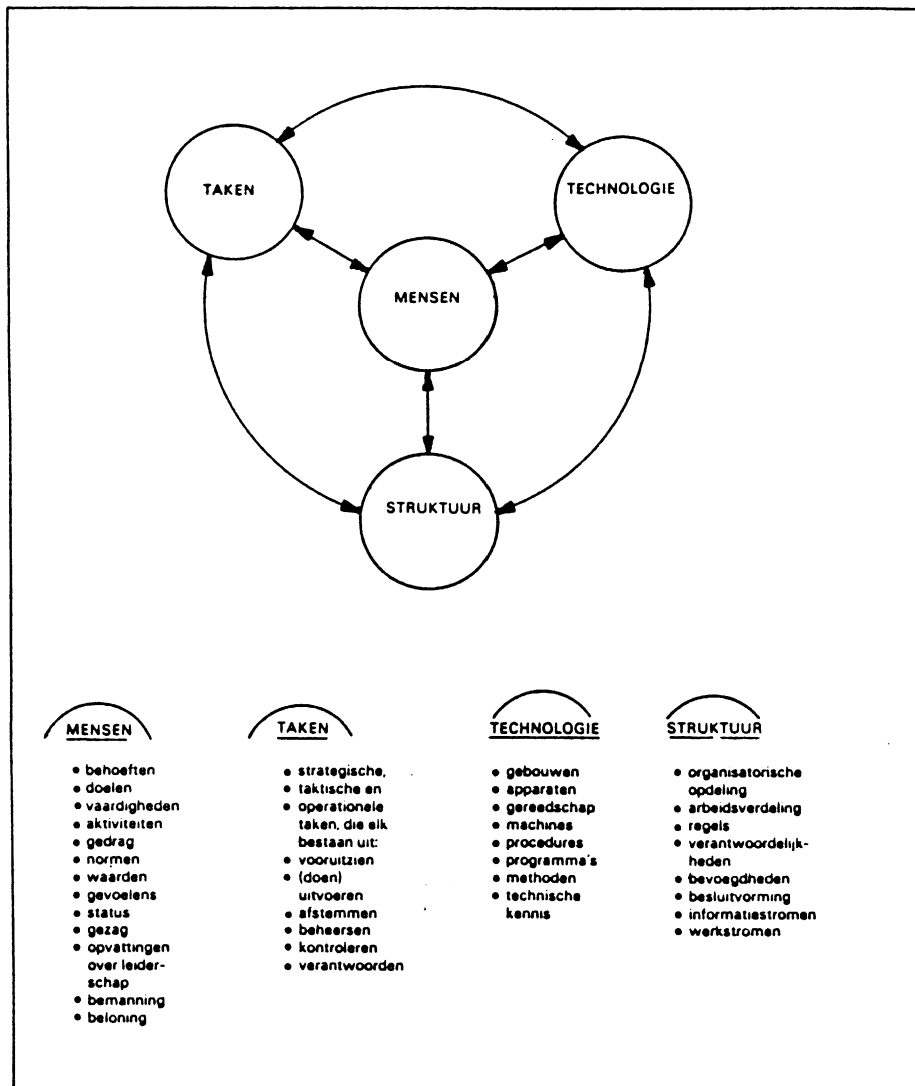
- de eisen die deze aanpak stelt aan de uitgangspunten en principes van het te ontwikkelen geautomatiseerde informatiesysteem: het STAA-produkt;
- de eisen die deze aanpak stelt aan het ontwikkelingstraject: het STAA-proces.

Daarbij moet de term 'produkt' niet worden opgevat als een hard te definiëren, na te streven situatie die vooraf bekend is en achteraf kan worden getoetst. Met 'produkt' wordt méér een set eigenschappen bedoeld die kan worden toegekend aan de te bereiken situatie, mede ten gevolge van het te doorlopen proces.

3 HET STAA-PRODUKT

3.1 Organisatie en informatiesysteem

Informatiesystemen worden ontwikkeld voor organisaties. Sterker: ze zijn een onderdeel van or-



Figuur 2: De Organisatie

organisaties. In figuur 2 is geprobeerd de complexe werkelijkheid die een organisatie is, weer te geven in een eenvoudig schema. Dit schema is een variant op de modellen van organisaties die French, Bell en Khandwalla hanteren, [9] en [14], die op hún beurt waren geïnspireerd door Leavitt[18]. Omwille van de eenvoud en in het kader van dit artikel zijn de omgeving en de voortbrenging (in- en uitgaande geld-, goederen- en dienstenstroom) weggelaten.

Het organisatiebeeld van figuur 2 zegt (onder andere) dat een verandering binnen de ene cirkel automatisch veranderingen ten gevolge heeft voor de andere. Het is zo gezien dus onmogelijk, in de technologie te wijzigen (bijvoorbeeld door geautomatiseerde informatiesystemen te ontwikkelen) zonder wijzigingen voor de mensen die in een organisatie binnen een bepaalde structuur taken verrichten.

In de automatiseringspraktijk wordt dan ook terecht soms gewezen op de nuttigheid van het parallel uitvoeren van een organisatie-onderzoek[25] en de wis-

selwerking tussen organisatie en informatiesysteem[11].

Moeilijker is het, de ontwikkeling van een informatiesysteem óók te beschouwen als een gewenste, bestuurbare verandering in de organisatie. Een informatiesysteem alléén beschrijven door middel van 'technische specificaties' en 'organisatorische consequenties' geeft dan een té beperkt beeld van de gewenste, toekomstige, veranderde situatie.

Idealiter zou automatisering in het volgende kader moeten plaatsvinden. Vanuit een duidelijke visie op de rol van de mens in de organisatie (waarover in 3.2 méér), zijn gewenste taakvervulling en de structuur waarmee dit wordt mogelijk gemaakt, wordt een informatie-/automatiseringsbeleid geformuleerd. Via een 'mens-vriendelijke' methode voor informatiesysteemplanning, zoals bijvoorbeeld PRISMA[17], wordt vervolgens aan dat beleid gestalte gegeven door middel van het ontwikkelen van informatiesystemen. Tijdens dat ontwikkelingsproces worden

dan regelmatig organisatorische keuzes gemaakt uit de technologische mogelijkheden.

3.2 De mens in de organisatie

Ontwikkelingsmethoden zien de mens in de organisatie vaak slechts in één hoedanigheid: een produktiefactor en dus uitwisselbaar met apparatuur en programmatuur, als dit uit een oogpunt van doelmatigheid of doeltreffendheid gewenst lijkt.

Toch is er méér: wat mensen in hun werk motiveert, is niet alléén het (tegen betaling) zijn van een produktiemiddel.

'Zo profileert de mens in de organisatie zich in drie rollen:

- als uitvoerder van een konkrete taak
- als behartiger van de eigen belangen
- als burger met eigen normen en waarden.

Als *uitvoerder* van een *taak* is de mens in de organisatie onderdeel van het hele middelencomplex waarmee het specifieke produkt of de dienst van die organisatie tot stand komt. Mensen opereren daar binnen het raam van de gemaakte afspraken en de vastgestelde wijze van produceren. Het werken van mensen wordt in dit verband beoordeeld naar de mate waarin het bijdraagt aan de totstandkoming van het produkt of de dienst.

Als *behartiger* van de eigen *belangen* stelt de mens in de organisatie aan de orde welke opbrengsten hij ontvangt uit het werken in die organisatie en zal hij proberen daarin in voor hem gunstige zin bij te sturen. Het gaat daarbij niet alleen om materiële opbrengsten, maar zeker ook om immateriële zaken. Mensen hebben opvattingen over wat zij verwachten van de organisatie en maken die opvattingen kenbaar. Hetzij door er zich over uit te spreken, hetzij door hun gedrag.

Als *burger* met eigen *normen en waarden* bepaalt de mens in de organisatie op welke wijze hij zijn taak uitvoert, wat hij van zijn bazen verwacht, hoe hij met kollega's omgaat en in hoeverre hij zich houdt aan de voorschriften en regels van de organisatie[6b].

De organisatorische keuzes waarbinnen automatisering zich afspeelt, worden dus óók ingegeven door opvattingen, normen en waarden van mensen. Dat dit kiezen enkel rationeel plaatsvindt, is een fictie[10]. Ontwikkelingsmethoden die de mens slechts zien als de rationele uitvoerder van een konkrete taak, komen tot een produkt dat die mens-als-geheel (lees: managers en gebruikers) niet past.

Een zienswijze die minder vreemd is dan zij sommige automatiseringsspecialisten en systeemontwikkelaars misschien toeschijnt: óók zij zijn mensen die in hun organisatie de drie genoemde rollen vervullen. (Wat dat betreft is bijvoorbeeld de term 'mensch-machinedialoog' verhullend: beter ware het te spreken van 'gebruiker-ontwikkelaardialoog'.)

3.3 Uitgangspunten

Wil men werken volgens STAA, dan moeten de betrokkenen gezamenlijk de volgende uitgangspunten aanvaarden: de optimale zelfbepaling en participatie van managers en gebruikers ten aanzien van hun toekomstige taakuitoefening (deze wordt in hoge mate beïnvloed door het te ontwikkelen systeem)

binnen de vanuit techniek én financiën gestelde minimumvoorwaarden en de minimaal noodzakelijke coördinatie van de wensen en verlangens van individuen/afdelingen.

De als laatste genoemde coördinatie is te bereiken, door de verbindingen te onderkennen tussen de afzonderlijk georganiseerde werkprocessen.

Deze verbindingen zijn de absolute eisen waaraan een ten behoeve van zo'n werkproces te ontwikkelen informatiesubstelsysteem voor andere subsystemen moet voldoen. Zolang men aan deze eisen voldoet, is men in de ontwikkeling van het produkt vrij, inclusief de daarbij te gebruiken (automatiserings)middelen. (Denk maar aan de filosofie van 'autonome groepen' in industriële omgevingen.)

De criteria bij de afweging van de uitgangspunten moeten dus niet alléén zijn gericht op de technisch-rationele en/of (ogenschijnlijk) laagste kosten, maar ook leiden tot bevredigende taken voor de betrokkenen.

Het zal duidelijk zijn, dat het hier gaat om het bereiken van een aanvaardbaar evenwicht tussen deze verschillende, maar tóch evenwaardige criteria.

De *konsequenties* hiervan moet men ten volle willen/kunnen dragen:-

- * Men moet bewust afzien van de doelstelling: het realiseren van een door enkelen - visionair - opgezette systeemconceptie.

Een gevolg hiervan kan zijn, dat er een discrepantie ontstaat tussen de aanwezige automatiseringsmogelijkheden en wat voor het produkt nodig is. Dit moet men bewust accepteren. Immers: de aanwezige technologie is geen alles bepalend uitgangspunt voor het nieuwe systeem, maar wél een faktor bij de afweging van de alternatieven.

- * Men moet investeringen aandurven, omdat - althans in de aanvangsfase van een socio-technische aanpak - doorlooptijd en tijdbesteding toenemen.

De winst van deze aanpak - waardoor de voornoemde investering wordt gerechtvaardigd - komt in latere fasen.

Men mag verwachten, dat een met wezenlijke participatie tot stand gebracht produkt 'beter' is. Het leidt tot grotere taakbevrediging en daardoor tot een - uiteindelijke - grotere effectiviteit in het gebruik van de automatiseringsmiddelen. De kans op teleurstellingen (geplande doorlooptijden die nooit worden gehaald, weinig acceptatie door de gebruikers van het nieuwe systeem, ogenschijnlijk onbeheerste, autonome stijging van de jaarkosten voor automatisering en dergelijke) wordt kleiner.

In de huidige automatiseringspraktijk is er - helaas - maar ál te vaak nóóit tijd om het goed te doen en - door de noodzaak afgedwongen - wél tijd om het over te doen.

- * Men moet aanvaarden, dat er 'extra' eisen worden gesteld aan het produkt, verband houdend met de daarmee samenhangende, toekomstige taakuitoefening van gebruikers. Deze eisen gaan vaak uit boven de klassieke eisen van gegevensinhoudelijke aard (welke gegevens leiden, na welke verwerking, tot welke informatie). Ze hebben (in

een veranderende organisatie) een dynamisch karakter.

3.4 Produkteisen

De bedoeling van het volgende overzicht is, aan te geven wat attentiepunten* zijn met betrekking tot de aan het produkt (het operationele informatiesysteem) te stellen 'extra' eisen: de eigenschappen die het eindresultaat zal moeten hebben.

De eerste twee groepen attentiepunten hebben vooral betrekking op het systeem-in-ontwikkeling.

De andere hebben meer te maken met de wisselwerking die er bestaat tussen het ontwikkelde systeem en de beleving van het werk.

Alle punten bevatten zaken die het meer of minder 'gelukt' zijn van een produkt kunnen bepalen. De mate waarin de genoemde punten moeten worden vervuld, is niet in algemene zin aan te geven. Bovendien zal nog aan de gebruikelijke criteria van efficiency, zingeving, doelmatigheid en overleving van de organisatie moeten worden voldaan.

De attentiepunten zijn als volgt gegroepeerd:

- | | | |
|--|---|--|
| <ul style="list-style-type: none">• Invloed tijdens de ontwikkeling en bij het gebruik van het systeem• Gebruikersgerichtheid van de invoer/uitvoer/informatie | } | Accent op het systeem-in-ontwikkeling |
| <ul style="list-style-type: none">• Vrijheid in het werk• Kreativiteit• Uitdaging en dynamiek• Vakkenis• Sociaal contact en samenwerking | } | Accent op de invloed van het systeem op het werk |

Invloed tijdens de ontwikkeling en bij het gebruik van het systeem.

Zowel tijdens de ontwikkeling van een systeem als het operationeel zijn van een systeem, dienen de (toekomstige) gebruikers invloed te kunnen uitoefenen. Dat betekent het een en ander. Het (te ontwikkelen) systeem mag niet onbedoeld tegen de bestaande machtsstructuren ingaan; ook mag het niet worden gebruikt als een voorwendsel om machtsstructuren te wijzigen of te bevestigen. Een grotere mate van decentralisatie van de informatievoorziening vergroot de mogelijkheid tot het uitoefenen van invloed door de gebruikers. De invloed van de gebruikers op de ontwikkeling van het systeem dient vooraf door afspraken te worden geregeld (er dient duidelijkheid te bestaan over de mate waarin waarden en normen tot hun recht kunnen komen). Tijdens de ontwikkeling van het systeem dient openheid te bestaan over de vaststelling van de toekomstige informatievoorziening (bespreekbaarheid informatiestromen). Het te ontwikkelen systeem mag de invloed van de gebruikers op hun dagelijkse werkuitoefening niet onaanvaardbaar verminderen. Inzicht in de werking van het systeem dient het blijvend invloed uitoefenen op het systeem mogelijk te maken. Openheid over de mogelijkheden van gebruik/misbruik van het systeem en het hanteren van een aantal spelregels bij

* Voor een uitgebreide toelichting op deze attentiepunten wordt de lezer verwezen naar de binnenkort te verschijnen KKC-monografie over dit onderwerp[16].

het hanteren van de informatie die het systeem oplevert is essentieel.

Gebruikersgerichtheid van de invoer/uitvoer/informatie

Gebruikersgerichtheid is essentieel voor de effectiviteit van een informatiesysteem. Veel hangt af van de acceptatie door de gebruikers van invoer en uitvoer.

De mate van acceptatie kan sterk worden beïnvloed door de wijze waarop invoer- en uitvoerdefinities tot stand zijn gekomen (mate van betrokkenheid en dergelijke). Daarnaast moet aandacht worden besteed aan: een 'gebruikersvriendelijke' invoerverzorging, een gebruikersgerichte uitvoer, een op de gebruikers afgestemde analyse- en dokumentatiemethode, alsmede ergonomisch verantwoorde en door de gebruikers geaccepteerde apparatuur, dialogen en gegevenspresentatie.

Vrijheid in het werk

Vrijheid in het werk is één van de met behulp van een STAA na te streven doelen. Het is een menselijke behoefte, te kunnen kiezen.

Deze vrijheid in het werk kan worden bevorderd indien: de status van het werk niet in negatieve zin wordt beïnvloed en er wordt voorkomen dat dagelijkse (operationele) beslissingen volledig uit het werkpakket verdwijnen. Ook moeten de gebruikers niet volledig afhankelijk worden van het geautomatiseerde systeem. Verder moet het systeem zo eenvoudig en begrijpelijk mogelijk worden opgezet.

Ook moet het systeem zodanig flexibel zijn, dat aan de (veranderende) wensen van de gebruikers kan worden voldaan. Tenslotte moet er een zodanige mate van decentralisatie aanwezig zijn dat gebruikers (-groepen) een bepaalde mate van autonomieit verkrijgen/behouden.

Kreativiteit

Het geautomatiseerde systeem zal van de gebruikers een zekere mate van creativiteit bij de uitoefening van het werk moeten eisen. Het is een menselijke behoefte, regelmatig *iets nieuws te doen*.

De volgende overwegingen spelen daarbij een rol: creativiteit is noodzakelijk om het geautomatiseerde systeem geen keurslijf voor de gebruikers te doen zijn; veranderingen van de aard van het werk moeten zodanig zijn, dat de nodige creativiteit niet wezenlijk vermindert; creativiteit is een belangrijke eigenschap van de mens, voor hemzelf én voor zijn organisatie.

Uitdaging en dynamiek

Om het geautomatiseerde systeem voor de gebruikers levend te houden, zal er een hoge mate van dynamiek van moeten uitgaan. Het is een menselijke behoefte, *uitdagend werk te doen*.

Dit is te bereiken door hoge eisen te stellen aan: de vermogens die nodig zijn om het werk te kunnen uitvoeren; de afwisseling die binnen het werk mogelijk is; de ruimte die wordt opengelaten voor het gezonde verstand en de intuïtie van de gebruikers.

Vakkennis

In een geautomatiseerd systeem dient de (in de organisatie aanwezige) vakkennis zo goed mogelijk te

worden gebruikt. Sterker nog, het beroep op de nodige (te verwerven) vakkennis dient, binnen redelijke grenzen, te worden gestimuleerd. Het is een menselijke behoefte, iets *te leren*.

Dit leren kan onder meer worden bevorderd door: het nadruk leggen op de automatisering als een ondersteunend hulpmiddel; het streven naar harmonieuze, zinvolle taken; het stimuleren van de reeds aanwezige individuele capaciteiten, kennis en ervaring.

Sociaal contact en samenwerking

Sociaal contact en samenwerking zijn van groot belang voor het goed functioneren van een geautomatiseerd systeem. Het is een menselijke behoefte, *iets en iemand te zijn*.

Dit houdt in, dat moet worden onderkend, dat de gebruikers in een sociaal systeem functioneren en dat zij moeten weten, wat er over de grenzen van het eigen werk heen gebeurt. Het systeem moet dus de ruimte bieden voor het bestaan van 'rationele' (formele) en 'irrationele' (informele) contacten tussen de gebruikers.

Daartoe moet reeds tijdens de ontwikkeling van het systeem de nadruk worden gelegd op de noodzakelijke samenwerking. Ook moet het systeem zodanig zijn opgezet, dat de individuele gebruikers inzicht hebben in de verbanden en samenwerkingspatronen binnen het systeem. Tenslotte moet veel aandacht worden besteed aan het functioneren van het individu binnen de groep, in het bijzonder waar het gaat om de verhouding tussen meerdere/ondergeschikte (gebruik van de informatie) en de loyaliteit van het individu ten aanzien van het systeem.

4 HET STAA-PROCES

4.1 Systeemontwikkeling is organisatie-ontwikkeling

Het 'ontwikkelen van een informatiesysteem' is in paragraaf 3 beschreven vanuit de bredere optiek van het 'veranderen van een organisatie'. De achterliggende ideeën over de mens in de organisatie en de uitgangspunten met betrekking tot het eindprodukt van het veranderingsproces zijn te kenschetsen door begrippen als 'zelfbepaling' en 'wezenlijke participatie'.

Naast de inhoud van het (te ontwikkelen) informatiesysteem, waarover in paragraaf 3 een en ander is gezegd, krijgt een socio-technische aanpak gestalte in de manier waarop de betrokkenen het ontwikkelings-/veranderingsproces inrichten.

Het ontwikkelingsproces moet evolueren van een multi-naar een interdisciplinaire aanpak. Dat wil zeggen, van een probleembenadering waarbij mensen van verschillende achtergronden/disciplines elkaar aanvullen naar een benadering waarin zij geïntegreerd, gezamenlijk en ten opzichte van elkaar nevenschikt werken. Hierbij denke men allereerst aan de betrokkenen van figuur 1 (specialisten, ontwikkelaars, managers en gebruikers), maar bijvoorbeeld ook aan personeelsfunctionarissen en organisatie-adviseurs.

Om dit te verwezenlijken moet aan een aantal voorwaarden zijn voldaan. De aspiraties moeten gelijk ge-

richt zijn. De individuele doelstellingen en belangen moeten over en weer bekend (of op zijn minst bespreekbaar) zijn. Elke 'discipline' moet over voldoende kennis beschikken ten aanzien van 'haar' aspect van de problematiek, waarbij de andere disciplines in die kennis vertrouwen moeten stellen (er is dus een gezamenlijke basiskennis nodig van automatisering én elkaars discipline). De samenwerking moet zijn gericht op het gebruik maken van alle aanwezige deskundigheden (waarbij de gebruiker de materiedeskundige bij uitstek is!).

4.2 Proceseisen

Vanuit de in het voorgaande vermelde aanpak, kunnen ten aanzien van het ontwikkelingsproces zelve, de navolgende eisen worden gesteld.

De samenwerking (binnen een projektorganisatie bijvoorbeeld) moet effectief zijn. Er moet dus een optimaal gebruik worden gemaakt van de aanwezige deskundigheden. Daarbij wordt de effectiviteit van de samenwerking bepaald door factoren als:

- samenstelling en bemanning van de projektorganisatie;
- aanwezige kennis met betrekking tot de ontwikkeling van het systeem;
- ervaringen met voorgaande samenwerkingsverbanden;
- onderlinge acceptatie van elkaars aspiraties, doelstellingen en specifieke kennis/ervaring.

Er moet een goede communicatie zijn zowel binnen de veranderingsorganisatie (projektorganisatie) als met de zogenaamde 'staande' organisatie. Verder moet worden bewaakt dat het te ontwikkelen systeem voldoet aan de (extra) eisen die daaraan worden gesteld. Om dit alles niet alleen te *willen*, maar ook te *kunnen* is een *leerproces* noodzakelijk, of zo men wil, een ontleren van wat elk vanuit zijn eigen vakgebied heeft ervaren aan 'feiten' en 'realiteiten'.

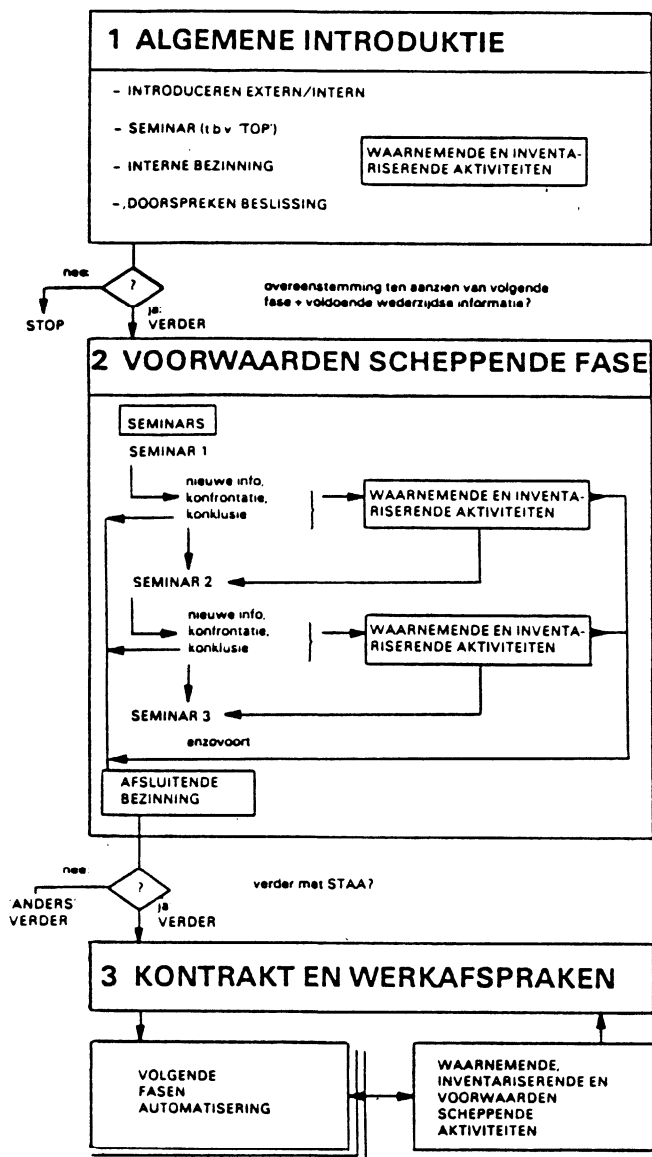
Dit leerproces is dan ook primair gericht op het onderkennen van en leren omgaan met de diverse (onderling verschillende) percepties van feiten en realiteiten. Daarbij wordt gewerkt aan het bereiken van overeenstemming over de mate waarin en de wijze waarop - rekening houdend met de feitelijke situatie binnen een organisatie en de factoren die daarin een rol spelen - de automatisering zal worden aangepakt. Het gaat daarbij om de (aanvaardbare) vrijheidsgraden bij de ontwikkeling van het informatiesysteem, de kaders, gezien de situatie waarin een organisatie verkeert (de feiten en realiteiten) en de opvattingen ten aanzien van de taakuitoefening en de zelfontplooiing van de leden van die organisatie.

4.3 De eerste stappen

De eerste stappen van een STAA zijn van wezenlijk belang voor het verdere proces: of de in het voorgaande bedoelde wezenlijke participatie wenselijk en mogelijk is, zal hier moeten blijken. In dit artikel ligt daarom de aandacht op de eerste, alles bepalende fasen van het STAA-proces.

Een en ander laat zich schematisch voorstellen (figuur 3).

(Varianten op dit raamwerk zijn overigens in verschillende praktijksituaties zeer goed denkbaar.)



Figuur 3: Het STAA-proces

De fasen van figuur 3 hebben (niet toevallig) de elementen in zich van 'wat zoal typische fasen zijn van de organisatie-ontwikkeling: opbouw van een relatie: organisatie - change-agent, voorbereiding der organisatie-leden, groepswork, probleemstelling, uitwerken van het na te streven doel, opstelling van een plan tot verandering, het plan uitvoeren en verbeteren, integratie'[28].

In de beschrijving van de fasen worden met 'adviseur' de personen aangeduid (van binnen of buiten de organisatie) die het idee hebben dat een STAA voor die organisatie nuttig zou kunnen zijn.

Met 'kliënt' worden de in een organisatie bij automatisering betrokkenen bedoeld.

Algemene introductie (fase 1)

De algemene introductie kan (afhankelijk van de si-

tuatie bij de cliënt) bestaan uit introductiedag(en), introductieseminar(s), waarnemende en inventariserende activiteiten.

In principe zal de algemene introductie een voor de betrokkenen min of meer vrijblijvend karakter hebben. In de algemene introductie dient de cliënt een beeld te verkrijgen van de specifieke aspecten van een STAA. Bovendien dient de basis te worden gelegd voor de voorwaarden scheppende fase (onder andere door introductie van STAA bij een aantal betrokkenen).

Daarnaast moet uit de algemene introductie informatie ten behoeve van de adviseur (en cliënt) voortkomen ten behoeve van de start van de voorwaarden scheppende fase, met name voor de inhoud van de in die fase te houden seminars.

Een belangrijk onderdeel van de algemene introductie wordt gevormd door de introductiedag(en). Het doel van de introductiedag is, te peilen in hoeverre men bereid is zich verder in de STAA-konceptie te verdiepen.

Met betrekking tot de introductiedag zijn de volgende opmerkingen te maken.

Het moet duidelijk zijn dat de introductiedag pas goed tot zijn recht komt in combinatie met het daarop volgende seminar.

De 'toon' van de introductiedag moet vooral zodanig zijn, dat de deelnemers (specialisten, ontwikkelaars, managers, gebruikers) aan deze dag zichzelf de vraag gaan stellen waarom de huidige aanpakken falen: vanuit het (h)erkennen van de eigen, ongewenste situatie, kan men konstruktief gaan denken over een STAA.

De introductiedagen zijn bedoeld voor topmanagement en andere sleutelfunctionarissen zoals 'de' automatiseringsman (of -vrouw).

Voorwaarden scheppende fase (fase 2)

De voorwaarden scheppende fase start na verkregen overeenstemming met de cliënt als vervolg op de algemene introductie.

Het begin van deze fase is niet vrijblijvend. De cliënt en de adviseur dienen te beseffen, dat in deze fase een veranderingsproces in werking wordt gezet, dat gevolgen voor de organisatie zal hebben, los van de vraag of men na deze fase wel of niet zal willen doorgaan. Na het doorlopen van de voorwaarden scheppende fase kan men in wezen niet meer terug. Indien deze fase goed is doorlopen, zal er 'iets' zijn losgemaakt. De daarop volgende vraag zal meer zijn: 'Hoe nu verder?', dan 'Wel of niet verder?'.

De doelstellingen van deze fase zijn:

- de algemene voorwaarden scheppen ten behoeve van het mogelijk maken van automatisering volgens 'STAA-filosofie',
- het '(her)definiëren' van het informatiebeleid, het automatiseringsplan en de (lopende) automatiseringsprojecten.

De voorwaarden scheppende fase start vanuit een situatie waarin de adviseur een zeker beeld moet hebben van de cliëntorganisatie (informatie uit de algemene introductie) en waarin een deel van de cliëntorganisatie (onder andere cliënt-top, personeelszaken, ondernemingsraad, automatise-

ringsafdeling) een zeker beeld heeft van aard en konsekventies van een STAA. Deze beelden worden aangescerpt door verdere waarnemende en inventariserende activiteiten én door enkele seminars waarin een groter deel van de cliëntorganisatie zich op de verdere aanpak bezint. (Om de gedachten te bepalen: een seminar duurt ongeveer drie dagen; het aantal deelnemers ligt tussen de 15 en 20.)

Struktuur en inhoud van deze fase zullen dus afhankelijk zijn van de aard, omvang en toestand van de cliëntorganisatie: in een kleine organisatie is misschien één seminar voldoende.

De kracht van STAA zit onder meer in de inbreng die in de seminars van de voorwaarden scheppende fase wordt geleverd vanuit de waarnemende en inventariserende activiteiten én in de inbreng die vanuit het ene seminar wordt gegeven aan het volgende. Seminars waarin onder andere wordt gewerkt aan het uitwisselen van de onderlinge percepties inzake automatisering.

Ook de konfrontaties tussen de waarnemingen van de adviseur en de konklusies van de deelnemers aan de seminars vormen een belangrijk onderdeel van deze fase. Zij zijn een voorwaarde voor het in 4.2 genoemde leerproces.

De parallel aan de seminars uitgevoerde waarnemende en inventariserende activiteiten betreffen de aandachtsgebieden: organisatie, ontwikkelingsniveau, automatisering, adviesverleden en het STAA-proces zélf. Daarbij wordt per aandachtsgebied materiaal verzameld over de 'objektieve' feiten, maar ook de opvattingen en attitudes, het organisatiegedrag en de wijze waarop de organisatie omgaat met veranderingsprocessen.

Voor een uitgebreide beschrijving van de seminars, de waarnemende en inventariserende activiteiten, zie [16].

Kontrakt en werkafspraken (fase 3)

Het kontrakt

De voorwaarden scheppende fase moet resulteren in een kontrakt en werkafspraken. Hieronder wordt verstaan, dat cliënt en adviseur een zodanig duidelijk beeld hebben verkregen van elkaar en van de konsekventies die een STAA voor de organisatie kan hebben, dat men bereid is met projectwerkzaamheden te beginnen. Dit houdt onder meer in, dat er duidelijkheid bestaat ten aanzien van het volgende:

- de aard/lokalisering van de problematiek;
- de minimumspecificaties ten aanzien van projecten: de proceseisen;
- de minimumspecificaties ten aanzien van de te ontwikkelen oplossingen: de produkteisen;
- de beperkende factoren waarmee moet worden rekening gehouden;
- de plaats van de automatiseringsprojecten in de participatie door staf en lijn);
- de te verwachten doorlooptijd en kosten;
- de kosten, verbonden aan eventuele funktieklassifikatie-bijstellingen ten gevolge van eventuele taakveranderingen;
- de wenselijkheid 'externe' partijen zoals vakbonden, leveranciers en afnemers te betrekken;
- de wenselijkheid een automatiseringsplan te

ontwikkeling dan wel, indien nodig, het bestaande plan bij te stellen;

- de gevolgen die een STAA kan hebben voor de organisatie.

Ook de (wellicht mede door een STAA geaccentueerde) onzekere factoren ten aanzien van aard van de oplossingen, gevolgen, kosten en doorlooptijd, dienen te worden onderkend. Dat wil zeggen: alle partijen moeten er zich van bewust zijn, dat deze onzekere factoren bestaan en dat men hiermee voor wat betreft STAA zal moeten leven.

Naast deze interpretatie van het kontrakt als 'wils-overeenkomst' zal er ook een kontrakt zijn, waarin de opdracht wordt geformaliseerd. Vorengenoemde punten worden daarin vastgelegd als overeenkomst tussen cliënt en adviseur.

Samenvattend: er moet worden bedacht, dat een STAA het accepteren van een zekere mate van onvoorspelbaarheid van de konkrete uitkomst van het projekt inhoudt.

De werkafspraken

Met de cliënt dienen werkafspraken te worden gemaakt, die voortvloeiën uit het formele kontraktdeel (probleemstelling, doorlooptijd, in te schakelen funktionarissen, enzovoort).

Deze werkafspraken zullen vooral zijn gericht op het vaststellen van: wie deelnemen aan de activiteiten, met welke bevoegdheid (inklusief het betrekken van de medewerkers en de plaats van de begeleiding); hoe dit geheel dient te funktioneren (qua planning, beheersing, rapportering, besluitvorming, werkmethode).

Op deze wijze kunnen ten behoeve van de eerste serie activiteiten afspraken worden gemaakt over wie wat, hoe, wanneer, waar en waarom doet en welke hulpmiddelen daarvoor kunnen worden aangewend.

Na afronding daarvan kan hetzelfde geschieden voor een volgende serie activiteiten waarmee het geheel een cyclisch aktie-karakter krijgt. Afhankelijk van de situatie kan dit geheel zich ontwikkelen binnen een grovere fasering die kontinu is te bewaken en bij te stellen door projekt- en/of stuurgroep. Funktioneert dit geheel goed, dan wordt een dynamische en beheerste afweging bereikt tussen voorspelbaar faseren (en daarmee 'in de greep houden') en situatie- en omstandigheidsgericht inspelen en ontwikkelen.

De cliënt

In het algemeen zal er sprake zijn van een formele cliënt (juridische opdrachtgever) én van een feitelijke cliënt.

Afspraken dienen plaats te vinden met de formele cliënt.

Echter, de overeenkomst om 'met elkaar in zee te gaan' dient te worden gebaseerd op een begrip bij de juridische opdrachtgever én de feitelijke cliënt over: de basiseigenschappen en konsekventies van een STAA, de nadere invulling en uitwerking van het veranderingsproces en de daarbij te hanteren projektorganisatie tijdens de ontwikkeling en implementatie. Het formele kontrakt regelt belangrijke voorwaarden scheppende afspraken en is te gebruiken als toetssteen

tijdens het veranderingsproces.

Daartoe dient het kontrakt in de verschillende stadia van het proces met de formele en met de feitelijke cliënt bespreekbaar te zijn en kan, indien de noodzaak daartoe bestaat, dit worden aangepast aan de gewijzigde situatie. Hieruit volgt, dat het kontrakt niet per definitie behoeft te worden beschouwd als een eenmalig vastgelegd iets, doch veel meer als een kader dat met het veranderingsproces kan meeëvolueren.

4.4 Het verdere verloop

Het in de beginfasen van een STAA afgesloten kontrakt fungeert als referentiekader voor de volgende fasen van de automatisering in een organisatie. Ruim gezien omvatten deze volgende fasen het gehele traject van het formuleren of bijstellen van het informatie-/automatiseringsbeleid via het op-/bijstellen van het automatiseringsplan, tot en met het uitvoeren van concrete projecten.

De daarbij nodige 'extra' activiteiten zijn van waarnemende, inventariserende en (zo nodig) van voorwaarden scheppende aard. Wat in de fasen 1 en 2 is begonnen, moet tijdens het verdere verloop worden voortgezet: de toetsingscriteria van het kontrakt en de werkafspraken geven 'extra' meetpunten in het verdere traject, evenals de produkteisen en de proceseisen.

In het (toch beperkte) kader van dit artikel wordt niet ingegaan op wat een en ander betekent voor:

- de relatie tussen informatie-/automatiseringsbeleid, -plannen en -projecten,
- de te hanteren methode(n) voor projectmanagement,
- de bemanning en inrichting van de veranderings-hulporganisatie (projektorganisatie) waarvoor figuur 2 óók geldt en
- de taken, rollen en verantwoordelijkheden voor de mogelijke partijen (automatiseringsspecialisten, systeemontwikkelaars, managers, gebruikers, personeelsfunctionarissen, organisatie-adviseurs, ondernemingsraad, vakbonden, leveranciers, afnemers van de organisatie, enzovoort).

De geïnteresseerde lezer zij verwezen naar de hiervoor genoemde monografie over STAA[16].

5 BRUG NAAR DE PRAKTIJK

In dit artikel is de huidige stand van zaken in de automatisering gekoncentreerd rond drie opgaven bij het ontwikkelen van systemen:

- een specificatie vanuit de technische én sociale dimensie van het informatiesysteem,
- een oplossing van de samenwerkingsproblemen bij de ontwikkeling en
- een voldoende besef van de situatie in een organisatie: de invloed van de aktualiteit.

Vervolgens zijn de kaders geschetst waarbinnen deze problematiek fundamenteel kan worden opgelost: een Socio-Technische Aanpak van Automatisering (STAA).

Het in de voorgaande paragrafen opgeroepen beeld van STAA is een ideaal beeld. De aanpak komt daar-

door misschien té zwaar over. De kritische lezer vraagt zich - ondanks het betoogde - wellicht tóch af: 'Waar zijn de concrete, meetbare resultaten?'

'Waar is al dat extra gepraat voor nodig?'

'Al die seminars, moet dat nu?'

Het antwoord ligt in het volgende.

In alle gevallen waar automatisering wezenlijk van invloed is op het bedrijfsgebeuren en een brede laag van medewerkers raakt, is een STAA te overwegen, als men méér realistische en beter acceptabele resultaten wenst. De (achteraf dure) missers worden vermeden; het klimaat voor automatisering wordt verbeterd.

In onze adviespraktijk lukt het - met vallen en opstaan - in een aantal situaties elementen van STAA met positieve uitkomsten toe te passen. In grote, complexe organisaties, waar gespecialiseerde stafafdelingen opereren vanuit onduidelijke rol-opvattingen. Maar ook in kleinschaligere omgevingen, waar het toch niet gemakkelijk blijkt, fenomenen als 'doelen', 'macht', 'waardering' en 'percepties' aan de orde te stellen. Het lukt overal waar men wil werken aan de drie bovengenoemde opgaven.

De brug naar de praktijk behoeft géén 'brug te ver' te zijn.

6 DANKWOORD

Mijn dank gaat uit naar allen die - direkt of indirect - aan de totstandkoming van deze publikatie meewerkten.

Met name moeten zij worden genoemd, die ooit mijn kollega's waren in de IOD-KKC-groep 'STAA'. Mijn medeleden van het STAA-netwerk en de NGI-sectie SAIA, werkgroep 1b, vormden een waardevol discussieforum.

Tenslotte dank ik u, volhardende lezer, die mij tot hier is gevolgd. Bedenk, dat de hele vorm van dit artikel (verbaal, stap-voor-stap) in strijd is met de titel. Je kunt nu eenmaal niet uitleggen hoe een sinaasappel smaakt.

Of zoals de Dodo tegen Alice zei: 'The best way to explain it is to do it'.

LITERATUUR

- [1] Bemelmans, T. M. A. en Boer, J. G. de: 'Het ontwikkelen van informatiesystemen'. INFORMATIE, jaargang 23, nr. 2, februari 1981. a: blz. 69. b: blz. 70.
- [2] Blokdijk, A.: 'Systeemontwikkelingsmethodiek SASO'. INFORMATIE, jaargang 23, nr. 4, april 1981. Blz. 223, 224.
- [3] Blokdijk, P.: 'Een analyse van de mogelijkheden voor het structureren van informatie systemen'. INFORMATIE, jaargang 23, nr. 5, mei 1981. Blz. 297 e.v.
- [4] Broek, J. G. A. van den: 'Het evalueren van methoden voor systeemontwikkeling'. INFORMATIE, jaargang 23, nr. 2, februari 1981. Blz. 77.

- [5] Dee, G. A. van, Ganzevoort, J. W. en Sonnenfeld, D.: 'Management-opleidingen - Liever maatwerk dan contekstie'. Handboek voor Managers - 3e editie (afl. 9). Blz. 7.3.0-102.
- [6] Dee, G. M. van, Jong, G. R. A. de, Kranendonk, A. en Verschuijl, M.: 'Management en Sociaal Beleid - De sociale dimensie bij de besturing van organisaties'. H. E. Stenfert Kroese B.V., 1981. a: blz. 20. b: blz. 26.
- [7] Drent, B. S., (Kranendonk, A. en Sonnenfeld, D.): 'De sociotechnische aanpak van automatiseringsvraagstukken'. Gidor-bulletin, oktober 1977, nr. 10. Blz. 17 e.v.
- [8] Edwards, Betty: 'Drawing on the Right Side of the Brain - A Course in Enhancing Creativity and Artistic Confidence'. J. P. Tarcher, Inc., 1979.
- [9] French, Wendell L. en Bell, Cecil H.: 'Organization Development - behavioral science interventions for organization improvement'. Second edition. Prentice-Hall, Inc., 1978. Blz. 40-42.
- [10] Ganzevoort, Wessel: 'Organisatie-ideologieën - De fictie van de rationaliteit van het organiseren'. Intermediair, jaargang 17, nr 36, 4 september 1981. Blz. 13.
- [11] Ganzevoort, J. W. en Heijden, A. van der: 'Organisatie en Informatie-ontwikkeling'. H. E. Stenfert Kroese B.V., 1980. Deel 1. Blz. 12.
- [12] Gelper, R. P. E.: 'Method/1'. INFORMATIE, jaargang 23, nr. 10, oktober 1981. Blz. 597.
- [13] Griethuysen, J. J. van: 'Grondslagen en terminologie voor het conceptuele schema'. INFORMATIE, jaargang 23, nr. 7/8, juli/augustus 1981. Blz. 471.
- [14] Khandwalla, Pradip N.: 'The design of Organizations'. Harcourt Brace Jovanovich, Inc., 1977. Blz. 227-235.
- [15] Koolhaas, J. W. en Visinescu-Voskamp, E. C.: 'Mogelijkheid tot beïnvloeding van de kwaliteit van arbeidsplaatsen door automatisering'. INFORMATIE, jaargang 23, nr. 23, maart 1981. Blz. 164.
- [16] Kranendonk, Aad en Sonnenfeld, Dezsö: 'Een kilo veren - mogelijkheden voor socio-technische automatisering'. KKC-monografie, najaar 1982.
- [17] Laagland, P. T. M. en Schaddelee, C.: 'PRISMA: een methode voor informatiesysteemplaning en systeemontwikkeling'. INFORMATIE, jaargang 24, nr. 1, januari 1982.
- [18] Leavitt, Harold: 'Applied Organizational Change in Industry' in: March, James (ed.): 'Handbook of Organizations'. Rand McNally, 1965.
- [19] Lokhorst, G. J. C.: 'Nobelprijzen 1981 - Geneeskunde/Fysiologie - Roger W. Sperry'. Intermediair, jaargang 17, nr. 50, 11 december 1981. Blz. 9.
- [20] Oudshoorn, H.: 'SDM en de technieken TIA, GOS, GOP en TOT'. INFORMATIE, jaargang 23, nr. 4, april 1981. Blz. 214.
- [21] Roest, W. F.: 'Het HOE en het WAT van het WAT en het HOE'. INFORMATIE, jaargang 23, nr. 9, september 1981. Blz. 538.
- [22] Ruys, H. D.: 'De ISAC-methodiek'. INFORMATIE, jaargang 23, nr. 5, mei 1981. Blz. 287.
- [23] Sebus, G. M. W.: 'Business Systems Planning'. INFORMATIE, jaargang 23, nr. 3, maart 1981. Blz. 143.
- [24] Sitter, L. U. de: 'Socio-techniek'. M & O, jaargang 25, nr. 2, maart/april 1974. Blz. 76.
- [25] Sloff, H.: 'Informatie, communicatie en organisatie; een projectbeschrijving'. INFORMATIE, jaargang 23, nr. 7/8, juli/augustus 1981. Blz. 431.
- [26] Sonnenfeld, D.: 'Wisselwerking tussen automatisering en modellen van organisatieverandering'. Presentatie KIVI, najaar 1977.
- [27] Tiggele-van de Wal, M. van en Snoo, H. C. R. de: 'Systeemrealisatie, theorie en praktijk'. INFORMATIE, jaargang 23, nr. 6, juni 1981. Blz. 360.
- [28] Vandenput, Michel A. E.: 'Organisatieontwikkeling: Kenmerken en verloop'. Mededelingen van het Verbond van Kristelijke Werkgevers (V.K.W.), januari 1971. Blz. 10, 11 en 12.
- [29] Vries, M. de: 'Informatiebehoeften zijn organisatiebehoeften'. INFORMATIE, jaargang 23, nr. 4, april 1981. Blz. 251.
- [30] Wilgen, J. van: 'Klassifikatie van organisatie en informatie'. INFORMATIE, jaargang 23, nr. 3, maart 1981. Blz. 154.

DE GEBRUIKER IN MODEL

Dr. H.G. Sol

Hoogleraar Ontwerpen van Informatiesystemen
Faculteit der Wiskunde en Informatica
Technische Universiteit Delft

Persoonlijk computergebruik leidt gemakkelijk tot een chaos van apparatuur, programmatuur, bestanden en modellen. Dit vraagt om een goede uitwerking van het informatiebeleid en de informatieplanning in organisaties.

De concretisering hiervan dient verder te gaan dan het aanstippen van de modewoorden vierde generatietalen, prototyping, beslissingsondersteuning en expert systemen onder de saus van persoonlijk computergebruik en informatiebeleid.

Informatiebeleid en informatieplanning dienen zich hier te richten op de ontwikkeling van een informatische infrastructuur waarin werkplekken worden ingericht voor taakuitoefening door individuen of groepen.

Het ontwerpen van werkplekken die een omgeving bieden voor de ondersteuning van de modellerende en de probleemoplossende taken van kenniswerkers, en het bedenken van een geschikte afstemming tussen deze werkplekken, vraagt om nieuwe technieken voor systeembeschrijving, systeemanalyse en systeemontwerp waarmee de modellerende gebruikers dynamisch in model worden gebracht.

1. INFORMATIEPLANNING EN PERSOONLIJK COMPUTERGEBRUIK

Ontwikkelingen rond informatisering in organisaties voltrekken zich in een hoog tempo. Generaties van computersystemen, van programmatuur, van formele talen en van ontwerpgereddschap volgen elkaar snel op.

Generaties gebruikers en ontwerpers hebben moeite de verschillende modegolven uit elkaar te houden en op hun waarde te schatten. In het licht van prototyping, management ondersteunende systemen, expert systemen en end-user computing is de roep om informatieplanning verklaarbaar. Het is echter de vraag of het hier om een tijdelijk verschijnsel gaat, en of het hier specifieke categorieën van informatieplanning betreft.

Moeten we bij alle modewoorden op onze hoede zijn, in het bijzonder end-user computing roept vraagtekens op. Allereerst is het niet op voorhand duidelijk wat we onder de eind-gebruiker moeten verstaan. Daarnaast blijkt dat de eindgebruiker moeilijk aan te wijzen is. Generaliseerbare conclusies m.b.t. categorieën gebruikers in relatie tot informatietechnologie zijn niet of nauwelijks te trekken, zie bijvoorbeeld Huber (1983) en Dickson (1986). Daarom willen we het liever hebben over persoonlijk computergebruik, waarmee we de behoefte aan informatieplanning in een historisch perspectief kunnen plaatsen.

Het gebruik van computers in organisaties kan men op verschillende wijzen indelen. In deze bijdrage onderscheiden we:

1. Geautomatiseerde gegevensverwerkende systemen (GGV),
2. Management informatiesystemen (MIS),
3. Persoonlijk computergebruik (PCG).

ad 1. Geautomatiseerde gegevensverwerkende systemen

Deze toepassingen worden gekenmerkt door de automatisering van massa-administraties. De ontwikkelingen worden beheerst door technische randvoorwaarden, met name beperkingen in rekensnelheid en toegankelijkheid van gegevens voor de opslag van gegevens.

De uitvoer van de geautomatiseerde processen is niet wezenlijk anders dan vóór de automatisering het geval is. De tijd, noodzakelijk voor de productie van data, kan worden ingekort. Daardoor kan in een aantal gevallen de frequentie van de informatievoorziening worden opgevoerd. Bij het ontwerpen van applicaties liggen de informatiebehoeften veelal vast. Centraal staat het volledig specificeren van de (administratieve) processen en de con-

structie van efficiënte programmatuur. De ondersteuning op het bestuurlijke vlak is gering, in het bijzonder bij de beslissingsvoorbereiding.

ad 2. Management informatiesystemen

In deze klasse van toepassingen wordt de computer meer en meer gebruikt voor planningdoeleinden. Van verschillende kanten, bijvoorbeeld vanuit de operationele research, worden algoritmen aangedragen waarmee planningproblemen op computers kunnen worden opgelost.

In het midden van de jaren zestig ontstaat het begrip management informatiesystemen. Door het merendeel van de auteurs wordt een MIS gedefinieerd, zie Burch en Strater (1974, p. 71), als een concreet systeem dat de volgende drie functies vervult:

- 'to meet legal and transactional data processing requirements,
- to provide information to management for support of planning, controlling, and decision making activities, and
- to provide a variety of reports, as required, to external constituents'.

Kenmerkend voor deze toepassingen is de veelal impliciet gehanteerde veronderstelling dat het functioneren van een organisatie kan worden beschreven door middel van gekoppelde modellen met verschillende niveaus van aggregatie die een totaal systeem vormen. Deze modellen bestaan meestal uit een stelsel vergelijkingen. De bruikbaarheid van deze modellen staat en valt met de mate waarin acceptabele verbanden tussen verklarende en verklaarde variabelen kunnen worden vastgelegd op basis van de beschikbare gegevens.

ad 3. Persoonlijk computer gebruik

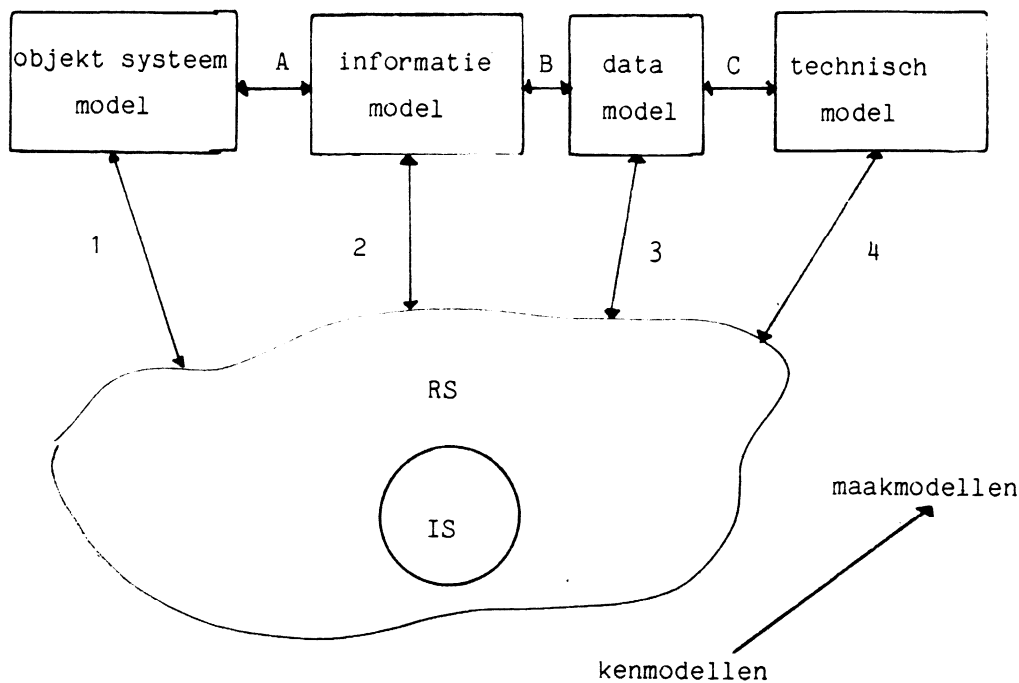
De derde klasse van toepassingen wordt mogelijk gemaakt door de ontwikkeling van database systemen en interactieve programmatuur. Door de mogelijkheden van gegevensbanken, mens-machine interactie, data-communicatie en tekstverwerking kan direct worden aangesloten bij individuele (of groepsgerichte) beslissings- en gegevensverwerkende processen. Deze processen worden veelal gekenmerkt door relatief weinig transacties, vaak met een hoge toegevoegde waarde en door een niet op voorhand duidelijk structureerbare probleemspecificatie, veelal met zachte gegevens, gericht op effectieve oplossingen.

Ontwerpen van informatiesystemen is een proces van probleemoplossen. Omdat

dit proces een onduidelijk gestructureerd en complex probleem is, wordt het vaak opgesplitst in hapklare brokken met een aanwijzbaar produkt.

De hapklare brokken hebben betrekking op duidelijk af te bakenen modelleringsstappen, die verschillende modellen opleveren. Allereerst zijn dit kenmodellen van het objektsysteem (OS), waarmee we de combinatie van reëel systeem (RS) en informatiesysteem (IS) die we beschouwen, aanduiden. Vervolgens komen we maakmodellen van tegen het te implementeren IS.

De samenhang hiertussen is geschetst in figuur 1.



Figuur 1. Modellen rond informatiesysteemontwerp

Ieder van deze modellen maakt onderdeel uit van een modellen-cyclus in een proces van probleemoplossen. Het ontwerpproces in figuur 1 levert een reeks van momentopnamen, waarbij ieder model een stadium op het ontwikkelingspad van kennen naar maken aangeeft.

Het complexe ontwerpprobleem kan men aanpakken door:

1. vanuit een bepaalde optiek tegen het probleem aan te kijken: Men gaat dan uit van een specifieke DENKWIJZE. Sommige ontwerpers nemen de bestaande bedrijfsfuncties als uitgangspunt en gaan na welke gemeenschappelijke datastructuren hierbij aansluiten. Andere beschouwen zeer gede-

tailleerd de bestaande processen in het objektsysteem.

2. het probleem in sub-problemen te verdelen. Men gaat dan uit van een specifieke AFBEELDINGSWIJZE.

Een voorbeeld is de indeling in:

- a. een systelogisch probleem, dat zich met de modellering van een objektsysteem vanuit een organisatorische invalshoek bezighoudt (het waarom),
- b. een infologisch probleem, waarbij de gegevensstructuren en de verwerkingsprocessen die nodig zijn om de gegevens te kunnen produceren centraal staan (het wat).
- c. een datalogisch probleem, dat zich richt op de wijze waarop en de vorm waarin het gegevensverwerkend systeem wordt gerealiseerd (het hoe),
- d. een technologisch probleem, waarin men moet aangeven welke technische hulpmiddelen worden gebruikt (het waarmee).

3. Een fasering in het ontwerpproces aan te brengen. Men let nu op de WERKWIJZE.

Davis e.a. (1980) stellen informatiebehoefte centraal in dit proces. Door een accurate en volledige beschrijving van de informatiebehoefte zal het informatiesysteem, met een grote kans op succes, ontworpen en geïmplementeerd kunnen worden.

Bij het bepalen van de informatiebehoefte bestaan twee belangrijke bronnen van onzekerheid. De eerste is het verkrijgen van een beschrijving van de informatiebehoefte. Bijvoorbeeld d.m.v. interviews, analyse van beslissingsprocessen, etc. De tweede is de garantie dat de verkregen informatiebehoefte akkuraat, consistent en volledig zijn.

Factoren die invloed hebben op de mate van onzekerheid zijn bijvoorbeeld definieerbaarheid van het probleem, structuur van bestaande systemen, projectomvang, begrip van de gebruiker en bekwaamheid van de ontwerper. Elke methodiek hanteert impliciet een strategie om onzekerheid te reduceren. Er worden wel vier strategieën onderscheiden, zie ook Sol en IJpelaar (1983):

- directe acceptatie van gebruikersbehoefte,
- een lineaire strategie,
- een iteratieve strategie,
- een incrementele of prototyping strategie.

Deze strategieën kan men terugvinden in de systemcyclus die bij het ontwerpen en bouwen van informatiesystemen wordt doorlopen.

4. Het projekt te besturen met de daarvoor beschikbare methoden en technieken. Nu staat de BEHEERSWIJZE centraal.

De beheersing van een projekt vraagt om aandacht voor tijd, geld en mankracht om een kwalitatief goed produkt op te leveren. De organisatie van het projekt en de informatie over de voortgang van het projekt zijn hierbij essentieel. Concreet gaat het om:

- de taakstelling voor het projekt en de daarin te onderscheiden activiteiten;
- de bevoegdheden van de projektleden;
- de volgorde planning van de activiteiten;
- de capaciteitsplanning van de te besteden menskracht en hulpmiddelen;
- de voortgangsbewaking;
- de vastlegging.

Een keuze voor denkwijze, afbeeldingswijze, werkwijze en beheerswijze vormen samen een projektaanpak, toegespitst op een specifieke probleemsituatie.

Voor de duidelijkheid zij gesteld dat denkwijze, afbeeldingswijze, werkwijze en beheerswijze zeker niet onafhankelijk van elkaar zijn.

In praktijksituaties kan voor het bepalen van een aanzet worden gegeven door aan de hand van tabel 1 een zogenaamde risico-analyse uit te voeren.

	post onvoorzien laag ————— hoog
probleem	
structuur	
omvang	
gebruikers	
ontwerpers	

Tabel 1.

Het gaat daarbij om het inschatten van het risico voor ieder van de factoren. Bijvoorbeeld, een duidelijk probleem (laag) in een afgebakende structuur (laag) van grote omvang (hoog) met onervaren gebruikers (hoog) en een redelijke ontwerpstaf (laag) vraagt om een lineaire strategie met accent op de infologie en de datalogie, terwijl de projektcontrole en gebruikersop-
leiding uitermate belangrijk zijn. Is het probleem onduidelijk (hoog) in een niet vastliggende structuur (hoog), doch van beperkte omvang (laag), dan staat het systemologisch probleem centraal, waarvoor een incrementele strategie gehanteerd kan worden, indien de gebruikers zich hierin kunnen vinden (matig) met gekwalificeerde ontwerpers (laag).

De fasen in de ontwikkeling van informatiesystemen zijn samenvattend in figuur 2 weergegeven, waarbij wij kijken naar de soort problemen die worden aangepakt, om naar de gevolgde denkwijze, afbeeldingswijze, werkwijze en beheerswijze.

FASE	GGV	MIS	PCG
soort problemen	(goed) gestructureerde processen	(goed) gedefinieerde informatie verzorging	slecht gestructureerde beslissingsondersteuning
denkwijze	procesgericht	gegevensgericht	objektgericht
afbeeldingswijze	deduktief, accent op hoe en waarmee	induktief accent op wat	hypothetisch-induktief, accent op waartoe
werkwijze	lineair	iteratief	incrementeel
beheerswijze	projektgroep	harmonie	participatief

Figuur 2

2. WRIJVINGSPUNTEN ROND PERSOONLIJK COMPUTERGEBRUIK

Het is opvallend dat de etiketten Persoonlijk Computer Gebruik (PCG) en Management Ondersteunende Systemen (MOS) te pas en te onpas worden gebruikt, zonder dat deze begrippen nauwkeurig worden omschreven. Vele schrijvers lijken de samenvattende term Decision Support Systems (DSS) te beschouwen als een aanpak waarbij dankzij moderne technologie individuele opvattingen en oordelen naar voren kunnen komen.

Klein en Hirschheim (1985) stellen dat 'there appears to be an implicit assumption on the part of DSS writers that DSS are beneficial to organizations and the DSS intervention process is not inherently polemic'.

Ginzberg en Stohr (1982) merken op dat de basis voor het definiëren van MOS is verschoven van een beschrijving van wat een MOS doet naar ideeën over hoe MOS kunnen worden ontwikkeld.

Deze verschuiving door de jaren heen kan als volgt worden toegelicht:

1. In het begin van de jaren '70 werd een MOS omschreven als een gecompute-

riseerd systeem om het nemen van beslissingen te ondersteunen. De basis voor PCG was gelegd in het toepassen van interactieve technieken en data bases om besluitvormingsprocessen te verbeteren. In dit MOS-concept lag een sterk cognitief gericht accent op individuele beslissers.

2. In de tweede helft van de jaren '70 legde de MOS-beweging de nadruk op 'interactive computer-based systems which help decision-makers utilise data bases and models to solve ill-structured problems'. De nadruk ligt minder op het beslissingsproces dan op de ondersteuning van het persoonlijk computergebruik met hulpmiddelen om snel toepassingen te ontwikkelen.
3. Daarna begint de reclame karavaan rond MOS en PCG pas goed op gang te komen. Alle systemen die de effectiviteit van managers zouden moeten verhogen krijgen het label MOS opgeplakt. Vakgebieden als operationeel onderzoek en psychologie onderkennen de verkoopwaarde van dit etiket. Begrippen als informatiecentrum en prototyping worden in één adem genoemd met MOS en PCG.
4. Recentelijk verschuift de aandacht naar intelligente werkstations waar expertsystemen en documentsystemen persoonlijk computergebruik in een data communicatie-netwerk ondersteunen.
Elam et al. (1985) vragen derhalve om een nieuwe visie op MOS en PCG. Zij stellen voor het begrip te beperken tot 'the exploitation of intellectual and computer-related technologies to improve creativity in decisions that really matter'.

De discussie rond de begrippen MOS en PCG dient niet de aandacht af te leiden van de vraag wat de bijdrage hiervan is om het reilen en zeilen van organisaties te verbeteren. Juist hier kunnen verschillende wrijvingspunten gesignaleerd worden op basis van de opgedane wervaringen:

a. Kennen en maken.

MOS richten zich op problemen die slecht gestructureerd zijn. Het is echter opvallend hoe weinig aandacht wordt besteed aan de stappen in het proces van probleemoplossen. Veelal onderscheidt men hier de stappen probleemonderkenning, probleemformulering, probleemoplossing en implementatie. Vele MOS richten zich op de laatste twee stappen. Echter, het lijkt niet onaannemelijk dat de eerste stappen meer aandacht gaan krij-

gen, zie bijvoorbeeld Landry et al. (1985), Sol (1982). Immers, voordat men een probleem kan oplossen of een MOS of informatiesysteem kan bouwen moet men de probleemsituatie in zekere mate kennen. Natuurlijk komen er ontwerpen of oplossingen tot stand, onder harde tijdsdruk of financiële druk. Dit wordt zeker in de hand gewerkt door de 'snel even doen' mentaliteit rond PCG. Toch is de stelling 'bezint eer ge begint' ook van toepassing op PCG en de ontwikkeling van MOS. Daarom dienen de probleemonderkenning en de probleemformulering meer aandacht te krijgen.

b. Aggregatie.

Bij het beschikbaar stellen van faciliteiten voor PCG gaat men vaak impliciet uit van de veronderstelling dat meer en betere informatie ook tot betere beslissingen leidt. Bovendien hanteert men veelal samenvattende of ingedikte gegevens over processen in organisaties.

In de praktijk van verschillende organisaties wordt onderscheid gemaakt tussen:

- een data base met (gedetailleerde) gegevens over het verloop van de primaire processen in de organisatie,
- een data base voor PCG met persoonlijke gegevens, externe gegevens en geaggregeerde gegevens uit de administratieve en transaktieverwerkende systemen.

Er zijn verschillende, voornamelijk technische, argumenten voor dit onderscheid:

- de efficiency van bestaande data base management systemen, en speciaal relationele DBMS, levert nog steeds technische problemen op wanneer men fysiek één data base probeert te realiseren.
- PCG vraagt om een andere omgeving met programmatuur en apparatuur dan transaktieverwerking.
- gegevens in de PCG-data base hebben een andere graad van nauwkeurigheid dan gegevens uit de transaktieverwerkende systemen.

De PCG-data base bevat veelal gegevens die via aggregatie tot stand gekomen zijn uit de onderliggende cijfers door bijvoorbeeld sommatie over tijd, over geografische gebieden, over produktgroepen, etc. Echter, het gebruik van deze gegevens om beslissingen te ondersteunen kan gevaarlijk zijn. Men moet een vraagteken zetten achter de waarde van management informatie die door aggregatie tot stand komt, zie Reuijl (1982), Sol (1983), Sol (1985). Het is niet gemakkelijk om in het algemeen een geschikte aggregatiegraad voor het nemen van beslissingen aan

te geven. De geaggregeerde cijfers zijn niet altijd geschikt om verbanden hard te maken die de werkelijkheid goed weergeven.

De conclusie hieruit is dat de technische beschikbaarstelling van bestanden voor PCG veel minder aandacht dient te krijgen dan de logische inrichting van deze bestanden. Deze conclusie dreigt echter verloren te gaan in het reclamegeweld rond relationele DBMS, 4de generatietalen, vraagtaalen en prototyping.

c. Spreadsheet modellen en discrete simulatie modellen.

De beschikbaarheid van gegevens en de bruikbaarheid voor het ondersteunen van beslissingen wordt veelal niet ter discussie gesteld. Men gaat er vanuit dat verbanden zoals vastgelegd in bijvoorbeeld spreadsheets de werkelijkheid goed weergeven.

Veel modellen die bij PCG gehanteerd worden, zijn z.g.n. vergelijkingenmodellen: een verzameling definitievergelijkingen en gedragsvergelijkingen. De gedragsvergelijkingen moeten worden geschat en gevalideerd. Daarna kan men met het model gaan spelen, bijvoorbeeld door 'what-if' vragen te stellen of door naar een optimale oplossing te zoeken. Echter, deze modellen zijn alleen toepasbaar onder de veronderstelling dat de vergelijkingen kunnen worden geschat en dat deze een goede beschrijving van de werkelijkheid geven.

De veronderstelling dat men een probleemsituatie kan beschrijven door in- en uitvoer van een 'black-box' te bestuderen, blijkt echter weinig op te gaan. Bovendien is een gebruiksvriendelijk pakket voor PCG geen waarborg dat goede modellen worden opgesteld.

Een oplossing kan worden gevonden door modellen te gebruiken die de black box openbreken. Met name in de discrete simulatie modellen worden de opeenvolgende fasen van een beslissingsproces wel vastgelegd. De fasen worden sequentieel doorlopen waarbij verschillende routes kunnen worden gevolgd. Deze modellen lenen zich beter voor het beschrijven van de verschillende fasen van beslissingsprocessen dan vergelijkingenmodellen. Verder zijn ze minder afhankelijk van de beschikbaarheid van gegevens dan vergelijkingenmodellen. Echter, discrete simulatie modellen vragen wel meer van het modelleringsvermogen van de gebruiker. Dit spanningsveld onderschrijft eens te meer dat een effectieve aanwending van faciliteiten voor PCG sterk afhangt van de modelleringskennis en de materiekkennis van de gebruiker.

d. Incrementeel ontwerpen.

Met betrekking tot strategieën voor het ontwerpen van informatiesystemen staat een incrementele werkwijze sterk in de belangstelling. Veelal wordt dit aangeduid met het modewoord prototyping. In het algemeen kan men van prototyping zeggen dat het zich richt op het snel creëren van werkende systemen. Daarmee kan worden nagegaan of het systeem een valide model is, dat aan de functionele eisen en wensen in de probleemsituatie beantwoordt.

Prototyping kan worden gehanteerd om:

- een beter begrip van een toepassingsgebied te krijgen (waartoe),
- informatiebehoeften juist en volledig te specificeren (wat),
- gebruikers-interfaces te ontwikkelen (hoe),
- (voor)productie testen uit te voeren (waarmee).

Op basis van onze ervaringen concluderen wij, zie Sol (1984b), dat bij prototyping:

- verificatie en validatie niet altijd de benodigde aandacht krijgen, doordat het snel hard maken van een dynamisch systeem de overhand heeft;
- niet iedere organisatie rijp is voor een wegwerp aanpak inherent aan prototyping. Met name geldt dit voor bureaucratisch getinte organisaties. Ook druist dit vaak in tegen de belevingswereld van 'traditionele' automatiseerders. Het over boord gooien van een oplossing waar tijd en mankracht in is gestoken, is niet altijd gemakkelijk voor de betrokkenen. Een organisatie moet voorbereid zijn op het naar boven komen van conflicten en bereid zijn tot het zoeken naar oplossingen hiervoor. Dit betekent dat (top)management een prototyping aanpak zwaar moet (onder)steunen;
- de oplossing snel als definitief door de organisatie wordt overgenomen zonder dat verder geëxperimenteerd wordt;
- modellen als vereenvoudiging van de uitgangssituatie gehanteerd worden, waarbij het totaalbeeld gauw uit het oog wordt verloren.
- eilanden van automatisering ontstaan.

Kortom, incrementeel ontwerpen staat niet garant voor een snelle convergentie naar een acceptabele oplossing.

Juist een incrementele aanpak rond PCG vraagt om een duidelijk kader in de vorm van een informatieplan en/of bedrijfsmodel.

e. Beheersing.

Diverse bedrijven stimuleren PCG om in te spelen op de mogelijkheden die de moderne informatietechnologie biedt. Vaak worden dan PCG en gebruik van micro-computers in een adem genoemd, niet weinig onder het argument dat de drempel voor acceptatie en gebruik verlaagd moet worden.

Wanneer men deze weg echter ongecontroleerd inslaat, kan dit gemakkelijk tot een chaos van programmatuur, apparatuur, persoonlijke bestanden en modellen leiden. Juist PCG vraagt om een aanpak, waarin naast het verlagen van de drempelvrees, voorwaarden voor een gecontroleerde en beheerste groei gerealiseerd kunnen worden.

f. Organisatie.

PCG kan niet los gezien worden van mogelijkheden die moderne informatietechnologie biedt om decentralisatie in de besluitvorming en/of in de informatievoorziening te realiseren. Dit is zelfs zover doorgegaan, dat PCG en decentralisatie in één adem genoemd worden. Er dient echter duidelijk onderscheid gemaakt te worden tussen besluitvorming, beheer van informatiesystemen en beheer van faciliteiten, die ieder in meerdere of mindere mate gecentraliseerd kunnen zijn.

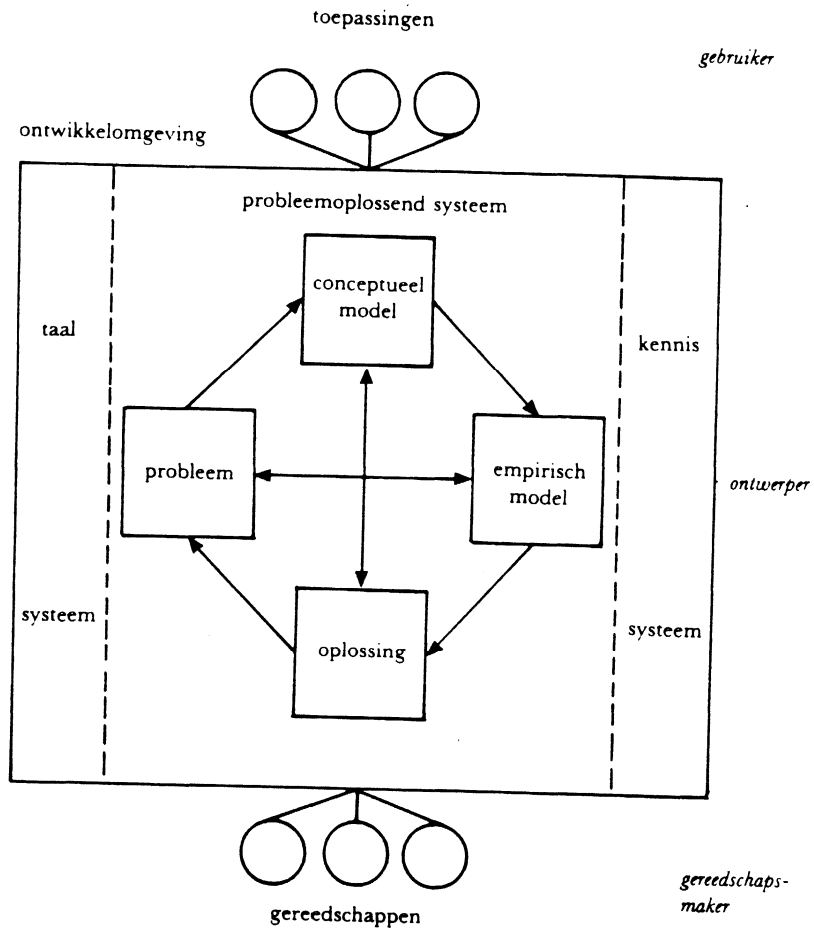
Vele MOS gaan stilzwijgend uit van de veronderstelling dat zwak gekoppelde systemen, bestaande uit individuen, groepen of afdelingen, een goed antwoord geven op de coördinatieproblemen in organisaties. Dit is echter helemaal niet vanzelfsprekend, zie bijvoorbeeld Bosman (1983).

Juist de moderne informatietechnologie biedt extra vrijheidsgraden voor het structureren van organisaties. Het is daarom des te opvallender dat bij de inrichting van PCG elementen als taakstructurering en afstemming erg weinig aandacht krijgen.

Deze gesignaleerde wrijvingspunten brengen naar voren dat het verschijnsel PCG in een passend jasje gestoken moet worden. Dit jasje zou door informatiebeleid en informatieplanning geboden moeten worden. Het is echter de vraag, in hoeverre hier sprake kan zijn van confectiekleding. Huber (1983) toont aan dat PCG moeilijk te categoriseren is. Dickson (1986) meldt zeer tegenstrijdige ervaringen rond het gebruik van verschillende faciliteiten voor PCG.

3. EEN VISIE

De ontwikkeling van PCG in een organisatie vraagt om een duidelijke strategie. Voor de formulering hiervan hanteer ik de volgende vertrekpunten.



Figuur 3

a. PCG zal zich gaan manifesteren in netwerken van kennisverwerkende systemen.

Op de werkplek worden gedistribueerde databases en (data)communicatiefaciliteiten geïntegreerd met audio-visuele technologieën, expert ondersteuning en lerende faciliteiten.

De plaatsing van een individu of groep in een netwerk roept onmiddellijk vragen op:

1. hoe ziet de taakstructurering op de werkplek eruit?
2. hoe wordt de inhoudelijke afstemming van de werkplekken gerealiseerd?

Deze vragen kunnen alleen beantwoord worden door de gebruiker in de verschillende alternatieven te modelleren.

- b. PCG zal zich richten op alle fasen in het proces van probleemoplossing. Naast vraagtafen worden modelleringsfaciliteiten minstens zo belangrijk, niet alleen gericht op vergelijkingenmodellen, maar zeker ook op procesmodellen. De voeding van de modellen met data wordt ondergeschikt aan de probleemformulering en modelspecificatie.

Deze vertrekpunten doen ons naar PCG kijken als omgevingen voor beslissingsondersteuning en kennisverwerking in een infrastructuur voor kenniswerkers.

Een dergelijke omgeving bestaat uit een beschrijvingsapparaat (een taalsysteem), een beschrijving van een toepassingsgebied (een kennisbank) en aanwijzingen hoe een conceptueel model en een empirisch model ontwikkeld kunnen worden, zowel beschrijvend als voorschrijvend (een probleemoplossend systeem), zie figuur 2.

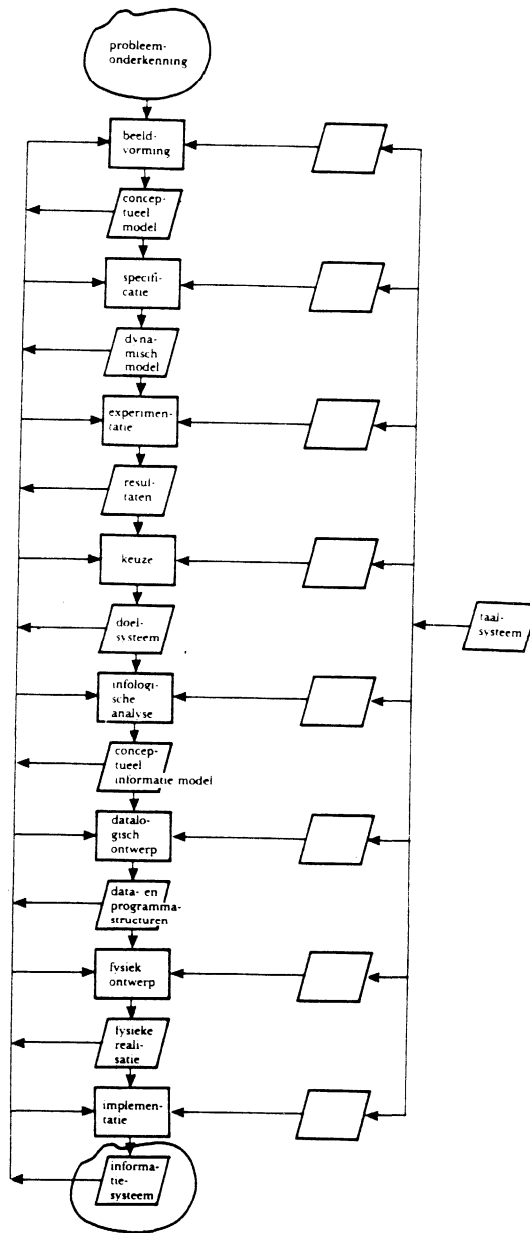
In deze omgevingen is, naar ik denk, een belangrijke plaats ingeruimd voor expertsystemen ter ondersteuning van het ontwerpproces. Ik denk daarbij aan de volgende verschijningsvormen, zie figuur 3.

1. Een expertstestem ter ondersteuning van de constructie van beschrijvende conceptuele modellen en empirische modellen van de bestaande situatie gebaseerd op 'application domain models'. In een dergelijk application domain zijn basisbegrippen en basisprocessen van een bedrijfstak of toepassingsgebied als prototype - in letterlijke betekenis - beschreven.
2. Een expertstestem ter ondersteuning van de constructie van een procesmodel door handvatten voor verificatie, validatie en gevoeligheidsanalyse aan te reiken.
3. Een expertstestem ter vergemakkelijking van de constructie van een proefopzet. Met nadruk zij opgemerkt dat 'automation of thinking in design' beperkt is, omdat deze hulp voor experimentatie en statistische analyses naar mijn mening slechts ten dele te automatiseren is.
4. Expertstestem die handreikingen kunnen leveren voor de overgang van kenmodel naar doelsysteem. Hier speelt echter de vraag in hoeverre zulke expertstestem generaliseerbaar zijn voor een specifiek toepassingsge-

bied en hoe deze expertise te verkrijgen is.

5. Een expertsysteem als kennisbank met mogelijke hardware- en softwareconfiguraties, die aan het maakmodel als uitkomst van het systemologische probleem, kunnen voldoen. Hiermee kan dan snel nagegaan worden of een systemologische oplossing infologisch, datalogisch en technologisch haalbaar is en tegen welke kosten.
6. Het lijkt mogelijk een kennisbank op te bouwen waarin empirisch materiaal over het verloop van automatiseringsprojecten is opgenomen, zoals kenmerken van de probleemsituatie, kenmerken van het produkt, en cijfers per activiteit, bijvoorbeeld projectduur en gebruikte middelen. Hieruit kunnen aanwijzingen voor de projectfasering geput worden.

Eén relativerende opmerking is echter rond deze expertsystemen op zijn plaats, en wel de valkuil van de expertise. De genoemde expertsystemen zijn bedoeld om tot betere oplossingen, ontwerpen en implementaties in toekomstige situaties te komen. De expertsie is en blijft echter gebaseerd op ervaringen uit het verleden. Een kritiekloos doortrekken naar de toekomst lijkt mij niet ongevaarlijk. Juist de combinatie van expertise en creativiteit moet informatiesystemen opleveren die de toets van falsificatie kunnen doorstaan.

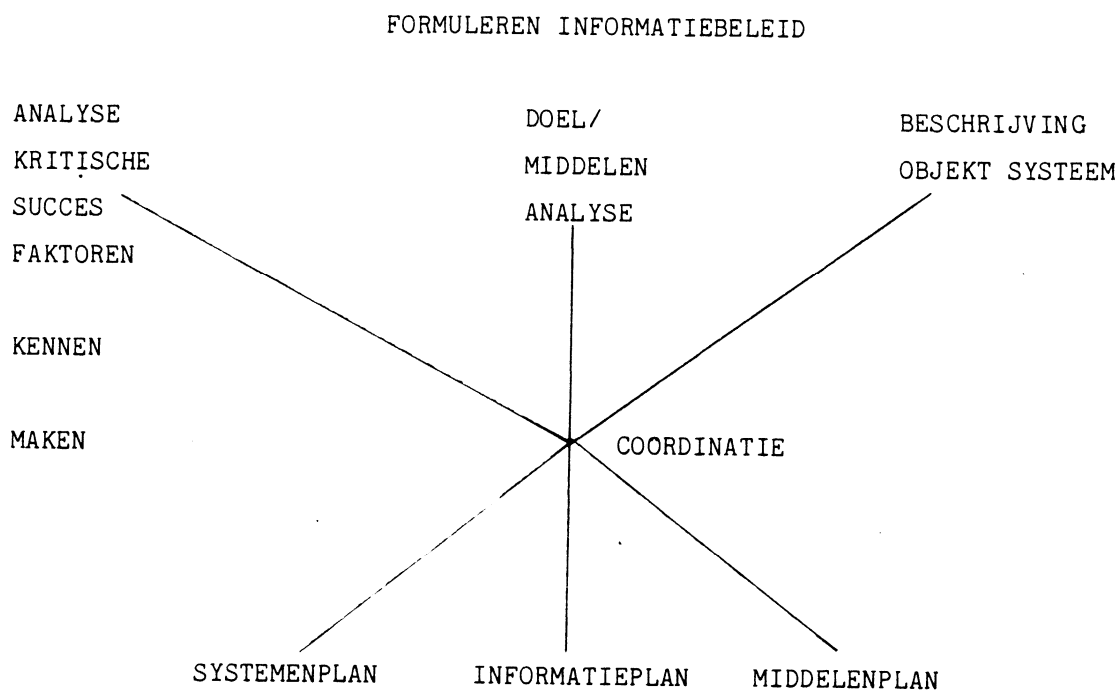


Figuur 4

4. GEVOLGEN

Welke zijn nu de gevolgen van de geschetste ontwikkelingen en specifiek deze invullingen van PCG voor de formulering van informatiebeleid- en informatieplanning?

Voor de beantwoording van deze vragen is een nadere aanduiding van informatiebeleid noodzakelijk. Belangrijke componenten bij de formulering van informatiebeleid zijn neergelegd in figuur 3.



Figuur 5

Deze figuur maakt duidelijk dat het resulterende PCG-plan onlosmakelijk

1. een INFORMATIEPLAN, omfattend de infrastructuur, waarin data-communicatiefaciliteiten zijn uitgewerkt tot op het niveau van de werkplekken en de opzet van de gegevensverzamelingen, zowel voor gemeenschappelijk gebruik als alleenstaand. Een belangrijk element hierin is het niveau van aggregatie van de gegevensverzamelingen en uitspraken over de relevantie van de inhoud hiervan voor het PCG.
2. een systeemplan, omfattend prioriteiten voor de te ontwikkelen en te

modificeren transaktieverwerkende systemen.

3. een middelenplan, waarin zowel centrale als de centrale apparatuur is aangegeven, alsmede richtlijnen voor standaardisatie en onderhoud en waarin projecten rond MIV en PCG m.b.t. PCG strategieën zijn aangegeven om drempels te verlagen en de acceptatie te verhogen.

Het informatiebeleid en de informatieplanning dient zich m.b.t. PCG te richten op de ontwikkeling van een informatische infrastructuur waarin werkplekken worden ingericht voor taakuitoefening door individuen of groepen. Een belangrijk aspect is hierbij de vraag in hoeverre de werkplekken van gemeenschappelijke gegevensverzamelingen gebruik maken.

Het ontwerpen van werkplekken die een omgeving bieden voor de ondersteuning van de modellerende en de probleemoplossende taken van kenniswerkers, en het bedenken van een geschikte afstemming tussen deze werkplekken vraagt om nieuwe technieken voor systeembeschrijving en systeemanalyse, waarmee de modellerende gebruikers dynamisch in model worden gebracht.

Een mogelijke aanpak is dan de volgende:

1. Maak een beschrijving van het objektsysteem.
2. Probeer vervolgens taken en beslissingsregels te achterhalen en maak hiermee een dynamisch model.
3. Analyseer de bestaande situatie met behulp van dit simulatiemodel en formuleer het probleem.
4. Creëer op basis van de kennis, opgedaan met dit kenmodel én op basis van een ontwerpfilosofie, alternatieve oplossingen en test deze als prototype.
5. Kijk hoe het prototype efficiënt in een informatieverwerkend systeem geïmplementeerd kan worden.

Een strategie zou dan kunnen zijn om per afdeling of groep van individuen een dynamische objektbeschrijving op een apart ontwikkelsysteem te realiseren als prototype. Vervolgens kunnen dan voor de organisatie of een gedeelte ervan de verschillende prototypes gecoördineerd worden. Op deze wijze kan de gebruiker spelend met modellen in model gebracht worden teneinde de mogelijkheden van de informatietechnologie te kunnen aanvaarden.

Dankwoord

De auteur dankt Prof. drs. B.K. Brussaard, Dr. H. Koppelaar en Drs. P.A.H.M. Mantelaers voor hun waardevolle en stimulerende opmerkingen over een concept voor deze bijdrage.

Referenties

- Alter, S.L., Decision Support Systems: Current practices and continuing challenges, Addison-Wesley, Reading, 1980.
- Bennet, J.L. (ed.), Building Decision Support Systems, Addison-Wesley, 1983.
- Bergquist, J.W., McLean, E.R., Integrated Data Analysis and Management Systems: An APL-Based Decision Support System, in: Sol, H.G. (ed.), Processes and Tools for Decision Support, North-Holland, Amsterdam, 1983.
- Bonczek, R.H., Holsapple, C.W., Whinston, A.B., Foundations of Decision Support Systems, Academic Press, 1981
- Bonczek, R.H., Holsapple, C.W., Whinston, A.B., Specification of Modeling and Knowledge in Decision Support Systems, in Sol, H.G. (ed.), Processes and Tools for Decision Support, North-Holland, Amsterdam, 1983.
- Bosman, A., Decision Support Systems: Problem Processing and Co-ordination, in: Sol, H.G. (ed.), Processes and Tools for Decision Support, North-Holland, Amsterdam, 1983.
- Bosman, A., Sol, H.G., Knowledge Representation and Information Systems Design, in Methlie, L.B., Sprague, R.H., (eds.), Knowledge Representation for Decision Support, North-Holland, Amsterdam, 1985.
- Burch, J.C. and Strater, F.R., Information Systems: Theory and Practice, Hamilton Publishing Company, Santa Barbara, 1974.
- Cats-Baril, W.L., Huber, G.P., DSS for Ill-Structured Problems: A Cognitive Approach and an Empirical Study, in: Methlie, L.B., Sprague, R.H. (eds.), Knowledge Representation for Decision Support, North-Holland, Amsterdam, 1985.
- Chang, C., Melamud, Y, and Seabrook, D., Expert Systems, The Butler Cox Foundation, Report Series no. 37, 1983.
- Dickson, G., Requisite Functions for a Management Support Facility, in: Sol, H.G. (ed.), Processes and Tools for Decision Support, North-Holland, Amsterdam, 1983.

- Dickson, G.W. et al., Understanding the Effectiveness of Computer Graphics for Decision Support: A Cumulative Experimental Approach, Communications of the ACM, January 1986, vol. 29, nr. 1.
- Elam, J. et al., A Vision for DSS, Proceedings DSS-85, 1985.
- Fick, G., Sprague, K.H. (eds.), Decision Support Systems: Issues and Challenges, Pergamon Press, Oxford, 1980.
- Fox, M.S., Knowledge Representation for Decision Support, Proceedings of the IFIP 8.3 Working Conference on Knowledge Representation for Decision Support Systems, Durham, 1984.
- Ginzberg, M.J. and Stohr, E.A., Decision Support Systems: Issues and Perspectives, in: Ginzberg, M.J., Reitman, W., Stohr, E.A. (eds.), Decision Support Systems, North-Holland, Amsterdam, 1982.
- Henderson, J.C., Ingraham, R.S., Prototyping for DSS: A Critical Appraisal, in: Ginzberg, M.J., Reitman, W., Stohr, e.a. (eds.), Decision Support Systems, North-Holland, Amsterdam, 1982.
- Huber, G., Cognitive Style as a Basis for MIS and DSS Designs: Much ado about Nothing?, Management Science, May, 1983.
- Humphreys, P., Lariche, O.I., Vari, A., Vecseny, J., Comparative Analysis of Use of Decision Support Systems in R & D Decisions, in: Sol, H.G. (ed.), Processes and Tools for Decision Support, North-Holland, Amsterdam, 1983.
- Keen, P.G.W., Adaptive Design for Decision Support Systems, Data Base, Vol. 12, No. 1 and 2, 1980.
- Klein, H.K., Hirschheim, R., Consequentialist Perspective of Decision Support Systems, Decision Support Systems, Vol. 1, no. 1, 1985.
- Klein, M., Manteau, A., Optrans: A Tool for Implementation of Decision Support Centers: in: Sol, H.G., (ed.), Processes and Tools for Decision Support, North-Holland, Amsterdam, 1983.
- Landry, M., Pascot, D., Briolat, D., Can DSS Evolve Without Changing Our View of the Concept of 'Problem'?, Decision Support Systems, Vol. 1, nr. 1, 1985.
- Lee, R.M., Epistemological Aspects of Knowledge-Based Decision Support Systems, in: Sol, H.G. (ed.), Processes and Tools for Decision Support, North-Holland, Amsterdam, 1983.
- Mitroff, J.J. et al., On Managing Science in the Systems Age: Two schemes for the study of science as a whole systems phenomena, TIMS Interfaces, vol. 4, nr. 3, 1974.

- Naylor, Th.H., Decision Support Systems or What Happened to MIS?, *Interfaces*, vol. 12, nr. 4, 1982.
- Olle, T.W., Sol, H.G., Tully, C.J. (eds.), *Information Systems Design Methodologies; A Feature Analysis*, North-Holland, Amsterdam, 1983.
- Olle, T.W., Sol, H.G., Verrijn Stuart, A.A. (eds.), *Information Systems Design Methodologies: A Comparative Review*, North-Holland, Amsterdam, 1982.
- Reuijl, J.C., *On the Determination of Advertising Effectiveness, An Empirical Study of the German Cigarette Market*, Stenfert Kroese, Leiden, 1982.
- Simon, H.A., *the New Science of Management Decision*, Harper and Brothers, New York, 1960.
- Sol, H.G., *Simulation in Information Systems Development*, Ph.D. Thesis University of Groningen, 1982.
- Sol, H.G., *Processes and Tools for Decision Support: Inferences for Future Developments*, North-Holland, Amsterdam, 1983.
- Sol, H.G., *The Emerging Role of Simulation Based Inquiry Systems*, in: Bemelmans, Th.M.A. (ed.), *Beyond Productivity, Information Systems Development for Organizational Effectiveness*, North-Holland, Amsterdam, 1984a.
- Sol, H.G., *Prototyping: A Methodological Assessment*, in: Budde, R. et al. (eds.), *Approaches to Prototyping*, Springer, Berlin, 1984b.
- Sol, H.G., *Aggregating Data for Decision Support*, *Decision Support Systems*, Vol. 1, no. 2, 1985.
- Sprague, R.H., *A Framework for Research on Decision Support Systems*, in: Fick, G., Sprague, R.H. (eds.), *Decision Support Systems: Issues and Challenges*, Pergamon Press, Oxford, 1980.
- Sprague, R.H., Carlson, E.D., *Building Effective Decision Support Systems*, Prentice Hall, Englewood Cliffs, 1982.
- Stamper, R., *Management Epistemology: Garbage In, Garbage Out*, in: Methlie, L.B., Sprague, R.H. (eds.), *Knowledge Representation for Decision Support*, North-Holland, Amsterdam, 1985.

