

Enkele gedachten over architectuur in de ICT

Jan L.G. Dietz
Technische Universiteit Delft
j.l.g.dietz@its.tudelft.nl

Een woord vooraf

Het is een slechte zaak dat er in het programma van LAC 2002 een track is opgenomen met de naam “Het Alternatief LAC”. Disputen en zelfs conflicten over wat architectuur in de wereld van de ICT is of zou moeten zijn, wat we er wel en niet mee aan moeten, zijn voorlopig gewoon onvermijdelijk en heel hard nodig. Daar leren we allemaal van. Ze zijn thans ook zeker niet van dien aard dat ze tot scheuringen in de ICT-gemeenschap zouden moeten leiden. Wat Jaap van Rees en Jaap Schekkerman met hun alternatieve LAC doen, is zich onttrekken aan de discussie, is hun gelijk bij voorbaat boven dat van anderen stellen, is althans de verdenking op zich laden dat ze niet willen luisteren naar wat anderen te vertellen hebben. Heeft het alom heersende virus van ongeduld en individualisme ook hen besmet? Is het alternatieve LAC een eenmalige ludieke wakkerschudder of houden we voortaan zo’n track in het programma? Hoe het ook zij en wat er ook van komt, goed is het niet wat er nu gebeurt.

Jaap van Rees kan best gelijk hebben als hij stelt dat de begrippen “architect” en “architectuur” binnen de ICT totaal verkeerd worden gebruikt en dat ze als vernieuwend worden gebracht juist daar waar er niets nieuws onder de zon is. Ook ik vind dat die begrippen vaak verkeerd worden gebruikt, maar ik zie ook uitzonderingen, en ik bespeur vooral veel onzekerheid bij degenen die de woorden in de mond nemen of uit hun pen laten vloeien. Laat de toekomst uitwijzen in hoeverre zij of ik of iemand anders met een mening over deze zaken gelijk heeft. En laten we tot die tijd met elkaar in discussie blijven, elkaar met argumenten bestrijden in plaats van laatdunkende opmerkingen rond te strooien over de ‘andersdenkenden’. Het vraagt geen moed je af te zonderen, demonstratief te protesteren, en alleen maar gelijkgestemden om je heen te verzamelen.

Wat ik in deze bijdrage aan het LAC wil doen, is niet anders dan wat ik de voorgaande jaren heb willen doen, namelijk de discussie gaande houden en die proberen netjes te voeren. Ik zou natuurlijk, net als Jaap van Rees, graag willen dat de lezer vindt dat ik steekhoudende argumenten heb en dat hij¹ zich er door laat overtuigen. In elk geval hoop ik dat mijn gedachten hem aan het denken zetten, dat hij voortaan kritischer zal zijn tegenover het gebruik van de termen “architect” en “architectuur”, door anderen en door hemzelf. Want dat is wel nodig (en daarin steun ik Jaap van Rees dus): dat we kritisch zijn en blijven.

1 Voorbereidende observaties en analyses

Tot ver in de jaren tachtig van de vorige eeuw leek het terrein van de ICT tamelijk duidelijk en daarmee tamelijk veilig voor ICT-ers². Wat men deed, was het ontwerpen, bouwen, implementeren en operationeel beheren van ICT-applicaties. De basisopvatting over de eigen professie in die periode ligt perfect besloten in dat ene woord *ICT-applicatie*. Wat het woord zo prachtig uitdrukt, is dat de ICT-er het als zijn taak zag ICT ten nutte te maken

¹ Zoals te doen gebruikelijk, dient men overal waar hij, zijn etc. staat ook zij, haar etc. te lezen.

² Met ICT-ers duid ik gemakshalve iedereen aan die werkzaam is in de ICT, in welke rol dan ook.

van de organisatie waar hij in dienst was of die hem had ingehuurd³. Laat ik gemakshalve alle (andere) belanghebbenden de organisatiemensen noemen. Deze organisatiemensen nu spraken liever van *informatiesystemen* omdat in dat woord hún bedoeling en het verwachte nut van de ICT-applicaties beter wordt uitgedrukt. Twee heel verschillende zienswijzen dus op hetzelfde fenomeen, niet per se onverenigbaar maar ook niet gemakkelijk te verenigen zoals iedereen weet die er bij betrokken was. Ondanks het feit dat het zeker geen sine cure was deze systemen te bouwen en operationeel te houden, was het werkterrein van de ICT-er wel overzichtelijk en duidelijk begrensd.

In de jaren negentig begint zich een proces van bewustwording af te spelen onder ICT-ers én organisatiemensen over de werkelijke reikwijdte en plaats van ICT. Dat proces van bewustwording werd op gang gebracht door de veranderende aard van de problemen waarvoor men zich gesteld zag. Van pure ICT-aangelegenheden werd de aandacht getrokken naar het (her)ontwerpen van bedrijfsprocessen, het management van bedrijfsprocessen, de innovatie van producten (met name in dienstverlenende bedrijven) en de noodzaak van intensieve samenwerking, tijdelijk of permanent) tussen bedrijven. Dat proces van bewustwording is nog niet beëindigd en de uitkomst dus nog ongewis, maar de richting is wel duidelijk. Die is dat men gaandeweg informatie- en communicatietechnologie steeds meer zal gaan zien als deel van een integrale *organisatietechnologie*. Daarmee bedoel ik het geheel aan middelen, en aan kennis omtrent de inzet van die middelen, om organisaties te laten werken, ze operationeel te laten zijn. Het omvat naast ICT bijvoorbeeld ook administratieve organisatie en personeelsmanagement. Want daar gaat het om: het beheerst, effectief en efficiënt veranderen van organisaties. Is dat allemaal nog wel een zaak voor de ICT-er? kan men zich terecht afvragen en het antwoord luidt: eigenlijk niet, dat wil zeggen, het is niet eigen aan de competentie van de ICT-er deze onderwerpen aan te pakken. Maar de bedrijfskundigen en organisatiekundigen blijken er niet beter voor geëquipeerd te zijn, en iemand moet het toch doen.

Voor wie de geschetste trend onderkent en begrijpt, is het niet moeilijk de problemen samen te vatten waar ICT-ers en organisatiemensen mee worstelen en waarvan beide partijen hopen dat het ‘tovermiddel’ architectuur de oplossing zal brengen:

- Om te overleven in het huidige, onrustige en steeds meer mondiale bedrijfsklimaat, is het noodzakelijk dat een bedrijf zijn productassortiment (goederen en diensten) snel en zonder veel kosten kan aanpassen.
- Om dezelfde reden kan het nodig zijn dat een bedrijf tijdelijk of permanent allianties aangaat met andere bedrijven. Denk aan ketenvorming in de logistiek en de zorgsector en aan de uitvoering van civiele bouwprojecten.
- De ontwikkeling en implementatie van grootschalige ICT-systemen is een haast niet meer te beheersen onderneming vanwege de vele aspecten waaraan aandacht moet worden geschonken en vanwege de vele belanghebbenden die er een rol bij spelen.
- Het gemakkelijk kunnen aanpassen van ICT-systemen aan nieuwe eisen en het kunnen hergebruiken van componenten zijn voornamelijk nog vrome wensen. Hetzelfde geldt voor het koppelen van systemen. Het is echter noodzakelijk dat die wensen in vervulling gaan.

De hulp die van het ‘tovermiddel’ architectuur wordt verwacht, is het leveren van inzicht in en overzicht over de complexiteit van de genoemde problemen, zodat ze beheersbaar en oplosbaar worden. De gangbare idee is, dat dat zou moeten kunnen door de hoofdzaken van iets vast te leggen en de details even terzijde te laten liggen. Maar is die verwachting wel gerechtvaardigd? En is dat dan architectuur? En los daarvan, is iets mogelijk om de

³ Voor de eenvoud van de uitleg neem ik menselijke organisaties als voorbeeld. De observaties, analyses en conclusies gelden echter ook voor ‘technische’ organisaties zoals embedded systems.

eenvoudige reden dat je het graag wil? Nee, natuurlijk niet, al zijn er altijd mensen die dat toch geloven en die het anderen willen doen geloven. Mijn hoofdboodschap in dit artikel is dan ook, dat er nog te weinig kennis is om de genoemde problemen écht te begrijpen en ze dus écht goed te kunnen aanpakken. Het ontwikkelen van kennis moet daarom prioriteit krijgen. Maar dat vraagt tijd en geld. De tijd is ons niet gegund, voorlopig zal de kennis over het toepassen van ICT in organisaties ver achter blijven lopen bij de potenties van de technologie (lees de hardware). Het geld is ons ook niet gegund, getuige de bedroevend lage onderzoeksfondsen die overheid en bedrijfsleven ervoor beschikbaar stellen. Ik kan daarom niet méér doen dan enkele leemtes in kennis aan te duiden en het begin van mogelijke opvullingen van die leemtes aan te geven.

2 Het wezen van architectuur

Het trekken van een parallel tussen architectuur in de ICT en architectuur in de bouwkunde ligt niet alleen voor de hand, we zijn het aan de bouwkundigen verplicht dat te doen, per slot van rekening hebben we het woord van hen geleend. Merkwaardigerwijs, in elk geval in tegenstelling tot wat je zou denken, valt het niet mee studies te vinden over het concept architectuur in de bouwwereld. Het lijkt iets zo vanzelfsprekends te zijn dat een discussie overbodig is. Er blijkt überhaupt weinig wetenschappelijk onderzoek te zijn gedaan naar het creatieve proces waarin een architect een ‘oplossing’ concipieert voor een ‘probleem’ dat hem wordt gesteld. Een bekende uitzondering is Christopher Alexander, die in de wereld van de ICT dan ook regelmatig wordt geciteerd. In dit artikel gebruik ik voor de afwisseling enkele meer recente uitspraken, en wel van de Nederlandse architect Wiek Röling. In zijn intreedende aan de TU Delft (1990), getiteld “De kunst van de ingenieur”, stelt Röling dat de kunst van de (bouwkundige) ingenieur is, ervoor te zorgen dat drie elementen volledig en in harmonie aan bod komen en elkaar versterken: functie (waarvoor?), materiaal (waarvan?) en techniek (hoe?). Een goed gebouw ziet er ‘vanzelfsprekend’ uit: het moet zo, en kan niet anders. Het geeft informatie over het hoe en het waarom. De inhoud (de functie) is aan de buitenkant af te lezen. Deze kwaliteiten maken een gebouw herkenbaar en eigen, geven het betekenis [einde citaat]. De kern van het begrip architectuur in de bouwkunde is dus de relatie tussen *functie* en *vorm* (waarbij vorm de combinatie is van materiaal en techniek). Die vorm is een ruimtelijke vorm, een bouwkundige architect schept fysieke ruimtes die een bepaalde functionele bedoeling hebben. Wat de ene architectuur beter of slechter doet zijn dan een andere, is de mate waarin de functie van een gebouw meer of minder direct blijkt uit zijn vorm, meer of minder ‘vanzelfsprekend’ is. De *onmiddellijkheid* (letterlijk: zonder tussenkomst van iets anders, dus zonder enige toegevoegde uitleg) van het zich manifesteren van de functie in de vorm is het ideaal van de architectuur in de bouwkunde.

Omdat fysieke vorm in de ICT niet bestaat (althans niet zo relevant is) is het concept architectuur niet zomaar over te nemen van de bouwkunde, maar wel via de omweg van de volgende redenering. Als de ene vorm beter past bij een nagestreefde functie dan de andere, gaat het blinkbaar om de vraag hoe een architect zijn ontwerpvrijheid gebruikt. Anders gezegd, als we het wezen van *architectuur* omschrijven als de *doelgerichte beperking van ontwerpvrijheid*, dan hebben we iets dat wel gemakkelijk en ook zinvol over te planten is naar de wereld van de ICT. Immers, het fysieke aspect van de bouwkundige vorm is nu geëlimineerd en in de toevoeging “doelgericht” ligt besloten dat men door het aanbrengen van de juiste beperkingen streeft naar een vorm die de beoogde functie ‘onmiddellijk’ realiseert, die haar ‘vanzelfsprekend’ doet zijn. Die vorm is nu een *informatische*, zoals ook de functie een informatische functie is. Daarmee bedoel ik dat er een informatische ruimte is (een Universe of Discourse) en dat er mogelijkheden zijn om in die ruimte te navigeren, d.w.z. om de informatie te vinden die je nodig hebt, en om te actualiseren, d.w.z. om informatie toe te voegen die je nodig vindt voor anderen of voor jezelf in de toekomst.

Deze omschrijving van architectuur biedt ook ruimte voor de inbreng van culturele waarden. In de bouwkunde vindt men daarvan vele voorbeelden. Ik citeer weer Wiek Röling: “De romaanse en gotische kerken ademen na eeuwen nog het godsvertrouwen (of de godsvrees) uit de tijd waarin ze gemaakt werden. In Zonnestraal⁴ weerspiegelt zich de optimistische hoop op een rechtvaardige samenleving.” Het culturele aspect hoeft overigens niet eens bewust bedoeld te zijn, het is gewoon onvermijdelijk. Het is immers uiteindelijk die éne mens, die niet alleen architect is maar ook lid van een culturele gemeenschap, die vanuit zijn heelheid een eigen weg zoekt en kiest in de onmetelijke ontwerpruimte. Dat we nog weinig ‘cultureels’ vinden aan ICT-systemen⁵ is, denk ik, vooral toe te schrijven aan het nog zo korte bestaan van het vakgebied.

Het beroep van bouwkundig architect is van oudsher op een ambachtelijk wijze bedreven en onderwezen. Men volgde het voorbeeld van de ‘meester’. Het is daarom niet verwonderlijk dat er in de loop der tijd diverse architectuurscholen of –stijlen zijn ontstaan. Het belang ervan voor de ICT zit hem daarin dat in zo’n school of stijl ervaring wordt vastgelegd, kennis wordt ontwikkeld. Door als onervaren architect binnen de ontwerpruimte die een stijl je oplegt te blijven, hoef je niet steeds weer dezelfde fouten te maken, niet steeds weer ‘het wiel uit te vinden’. Een *architectuurstijl* is dus een *stelsel van richtlijnen voor het ontwerpen*, richtlijnen voor het bewust en doordacht beperken van de ontwerpvrijheid. Meer prozaïsch gesteld, maar daarom niet minder waar: een architectuurstijl is een *standaard*.

3 Een werkdefinitie voor architectuur in de ICT

De ontwerpvrijheid van de ICT-architect is vele malen groter is dan die van de bouwkundige architect omdat de ICT-architect nauwelijks wordt gehinderd door de beperkingen van de ‘materialen’ die hij gebruikt. Bits zijn nu eenmaal bijzonder gewillig en plooibaar. Vanwege die immateriële aard van ICT-systemen, is het ‘ontwerpen onder architectuur’ voor de ICT des te meer geboden en zou men elke opgelegde zinvolle beperking van ontwerpvrijheid moeten verwelkomen en omarmen. De crux zit hem natuurlijk in de toevoeging ‘zinvolle’. Hoe bepaal je dat, ofwel wat is een voor de ICT zinvol en praktisch bruikbaar architectuurconcept? Ik denk dat dat in elk geval een concept dient te zijn waarin het idee van stijl een noodzakelijk onderdeel is. Alleen door te standaardiseren en te normaliseren kan de complexiteit van de hedendaagse problemen worden beheerst. Concreet dus zou architectuur in de ICT kunnen worden omschreven als *genormaliseerde doelgerichte beperking van ontwerpvrijheid*, waarbij met normaliseren is bedoeld het zich conformeren aan de architectuurstijl die als de standaard geldt of die als de standaard is gekozen.

Laten we op deze omschrijving voortbouwen, maar laat ik ook meteen twee praktische consequenties trekken. Ten eerste, architectuur is blijkbaar niet hetzelfde als ontwerp. Een ICT-architect kan dus niet bijvoorbeeld een of ander diagram tonen aan zijn opdrachtgever en zeggen: kijk dit is de architectuur van het systeem dat ik wil bouwen. Het voegt ook helemaal niets toe, niet aan de waarde van het ontwerp noch aan de status van het beroep, een tekening die je eerst ‘globaal ontwerp’ noemde nu ineens ‘architectuurmodel’ of zo te noemen. In dit verband deel ik volledig de kritiek van Jaap van Rees, zoals verwoord in de aankondiging van het alternatieve LAC, dat “... de begrippen ‘architectuur’ en ‘architect’ ... als vernieuwend gebracht (worden) daar waar er niks nieuws onder de zon is.” Ten tweede, omdat de architectuur van een systeem zich toont in de realisatie van het systeem, moet die architectuur al wel in het ontwerp besloten liggen. Dat is ook zo. Iemand die de ‘tekeningen’ van de ICT-

⁴ Zonnestraal is de naam van het bekende, door architect Duiker ontworpen, sanatorium

⁵ Ik gebruik voortaan de term ‘ICT-systeem’ als een compromis tussen ICT-applicatie en informatiesysteem

architect begrijpt, zal de architectuur (nogmaals: de genormaliseerde doelgerichte beperking van ontwerpvrijheid) uit de ‘ontwerptekening’ kunnen aflezen. Kortom, er is alle reden om de architectuur en het ontwerp van ICT-systemen strikt en consequent te scheiden.

Met name bij dynamische systemen (waartoe ICT-systemen behoren) is het zinvol twee architecturen te onderscheiden, namelijk functie-architectuur en constructie-architectuur. Met *functie-architectuur* bedoel ik de ‘uitwendige’ architectuur van een systeem, de architectuur van de informatische ruimte waartoe de gebruiker toegang heeft, inclusief de navigatie- en actualiseringsmogelijkheden daarin. De functie-architectuur van een ICT-systeem komt dus overeen met de (volledige) architectuur in de bouwkunde. Functie-architectuur toont zich in de concrete details van de gebruikersinterface (de vorm). In de functie ligt het mogelijke *gedrag* van het systeem, dat is de uitwendige manifestatie van het *systeem-in-operatie*, dus wat er uitwendig gebeurt als het systeem in bedrijf is, besloten. Het is de kunst van de ontwerper dat mogelijke gedrag onmiddellijk te laten blijken uit de details van de interface. Een naar mijn mening nog steeds onovertroffen staaltje van goede functie-architectuur is het Macintosh OS waarmee Apple een jaar of twintig geleden de wereld verraste. De metafoer van het bureaublad herstelde de gebruiker in zijn waarde (sic!). Hij hoefde niet meer de operator te spelen die een heleboel nogal cryptische commando’s feilloos moest kennen om de computer te laten doen wat hij wilde, hij kon dat nu gewoon direct met een simpele en intuïtieve ‘drag and drop’.

Met *constructie-architectuur* bedoel ik de ‘inwendige’ architectuur van een systeem, de architectuur die geldt voor de *compositie* (de verzameling componenten) en de *structuur* (de interactie tussen die componenten) van het systeem. In de constructie ligt de mogelijke *werking* van het systeem, dat is de inwendige manifestatie van het *systeem-in-operatie*, dus wat er inwendig gebeurt als het systeem in bedrijf is, besloten. Het is de kunst van de ontwerper die mogelijke werking onmiddellijk te laten blijken uit de details van de constructie. Wellicht ten overvloede, maar toch: de constructie-architectuur van een systeem is niet het globale ontwerp ervan, het is niet het ‘blokkendiagram’. Het is wel, en alleen maar, de stijl waaraan de ontwerper zich heeft geconformeerd bij het ontwerpen van de constructie. Voorbeelden van constructie-architectuur zijn de C/S-architectuur en de O-O-architectuur.

Uiteraard is er een verband tussen de constructie en de functie van een systeem. Dat verband is, dat de functie van een systeem wordt *verwerkelijkt* in de constructie van het systeem, met name in de constructie van de gebruikersinterface. Omgekeerd kan men stellen dat het gedrag van een systeem wordt *verklaard* door de werking van het (gehele) systeem. Het ‘onder architectuur’ hebben ontworpen van de constructie garandeert niet dat die constructie de functie op een correcte manier verwerkelijkt, het is wel zo dat daarmee het verifiëren van de constructie gemakkelijker wordt gemaakt. Iets soortgelijks gaat op voor de functie van een systeem. Het ‘onder architectuur’ hebben ontworpen van de functie garandeert niet dat die functie een correcte neerslag is van de functionele requirements, het is wel zo dat het valideren van de functie gemakkelijker wordt gemaakt.

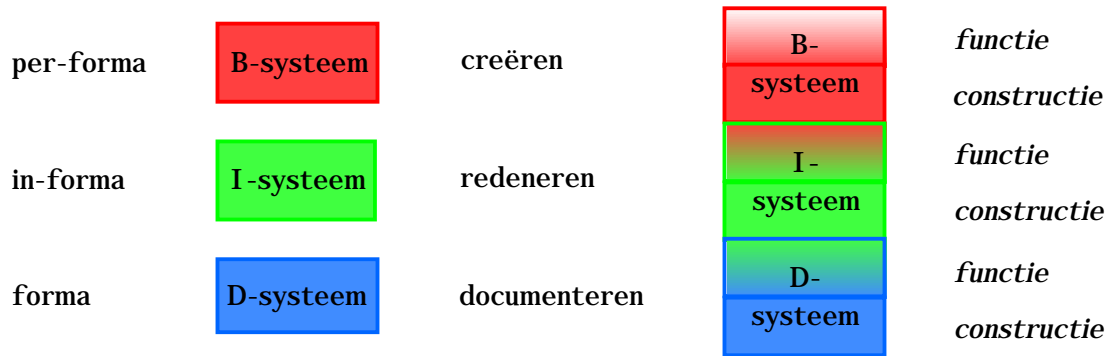
Samenvattend kom ik tot de onderstaande werkdefinities voor architectuur in de ICT:

Constructie-architectuur is de genormaliseerde beperking van ontwerpvrijheid bij het ontwerpen van de constructie van een systeem. Het ontwerp van een constructie is ideaal als de werking van het systeem onmiddellijk blijkt uit zijn compositie en structuur.

Functie-architectuur is de genormaliseerde beperking van ontwerpvrijheid bij het ontwerpen van de functie van een systeem. Het ontwerp van een functie is ideaal als het gedrag van het systeem onmiddellijk blijkt uit de constructie van de gebruikersinterface.

4 De ICT-gerelateerde aspectsystemen van een organisatie

ICT-systemen bieden, zoals we zagen, informatische ondersteuning aan menselijke actoren in organisaties. In de aard van die ondersteuning kunnen drie soorten worden onderscheiden. Anders gezegd, aan een organisatie kunnen vanuit de optiek van de ICT drie aspectsystemen worden onderscheiden, die in figuur 1 zijn uitgebeeld, en die zijn afgeleid van drie informatie-aspecten: het forma-, het informa- en het performa-aspect.



Figuur 1 De drie aspectsystemen van een organisatie

Met het *forma*-aspect wordt alles wat met de verschijningsvorm van informatie te maken heeft bedoeld. Het omvat de in de semiotiek onderscheiden niveaus empirie en syntaxis. Als men door de ‘forma-bril’ naar een organisatie kijkt ziet men menselijke actoren bezig zijn met *documenteren*, d.w.z. met het opbergen, verzamelen, transporteren en kopiëren van documenten. Deze actoren zijn de elementen van het D-systeem van de organisatie (de D van document en van data). Ze zijn bezig met informatie zonder zich te bekommeren om de inhoud ervan. Typische voorbeelden van D-actoren zijn de archivaris en de interne postbode. De inhoud van informatie komt juist wel in het *informa*-aspect aan de orde. Nu gaat het om wat er in de vorm is uitgedrukt, om de inhoud of de betekenis dus. Dat aspect omvat de niveaus semantiek en pragmatiek uit de semiotiek. Als men door de ‘informa-bril’ naar een organisatie kijkt ziet men menselijke actoren bezig zijn met *redeneren*, d.w.z. met het onthouden, zich herinneren en uitwisselen van kennis en met het berekenen van afgeleide kennis uit bestaande. Deze actoren zijn de elementen van het I-systeem van de organisatie (de I van informatie en intelligentie). In tegenstelling tot het D-systeem, is het I-systeem, vanwege de volledige abstractie van de vorm, puur conceptueel. De rol die de mens in het informa-aspect speelt, is die van rationele actor, van iemand die weet en redeneert. Typische voorbeelden van I-actoren zijn de boekhouder en de (markt)analist. Resteert het *performa*-aspect. Kijkend naar een organisatie door de ‘performa-bril’ ziet men menselijke actoren die nieuwe feiten creëren en daarmee de wereld veranderen. Het handelen van deze actoren kan materieel zijn (de fabricage of het transport van goederen) of immaterieel (het nemen van besluiten of het vellen van oordelen). Kenmerkend voor het creatieve handelen (en fundamenteel verschillend van rationeel handelen) is, dat het de enige vorm van handelen is die originele feiten tot stand brengt. Mensen in hun rol van creërende actor zijn de elementen van het B-systeem van een organisatie (de B van bedrijf en business). Typische voorbeelden van B-actoren zijn de verkoper en de inkoper.

Zoals statistieken liegen, zo kunnen ook figuren liegen, dat wil zeggen, je ziet er vaak niet alles in wat is bedoeld en je ziet er dingen in die niet zijn bedoeld. Dat geldt ook voor figuur 1 (en voor de andere figuren). Het is bijvoorbeeld wel zo dat de drie aspectsystemen aan elke organisatie zijn te onderscheiden, maar het is niet waar dat het verband tussen het I-systeem

en het D-systeem net zo is als dat tussen het B-systeem en het I-systeem, hetgeen de tekening wel suggereert. Iets wat ik wel bedoel te laten zien maar dat er lastig uit te halen is, is dat de drie aspectsystemen tezamen een compleet geheel vormen, d.w.z. dat ze noodzakelijk zijn en voldoende (er is geen vierde). Die eigenschap heb ik willen uitdrukken door de keuze van de drie primaire kleuren: rood, groen en blauw, die in de kleurenleer een complete set vormen.

De geschiedenis van de ICT laat de evolutie van het denken van ICT-ers zien, namelijk het achtereenvolgens ‘ontdekken’ en zich eigen maken van de drie besproken aspectsystemen (van beneden naar boven in figuur 1). Het begon in de jaren vijftig met het D-systeem. Papieren bestanden werden klakkeloos omgezet in een elektronische vorm. De belangrijkste bewerkingen waren het invoeren van data, het sorteren en samenvoegen van bestanden en het printen van data. Pas in de jaren zeventig ontstond er serieus aandacht voor de inhoud van die gegevens. De ‘infologische’ beweging, die door Langefors al eind jaren zestig was ingezet, leidde onder meer tot de ontwikkeling van databasemanagementsystemen en van de gestructureerde analyse- en ontwerpmethoden. Het besef drong door dat je ICT-systemen ‘eruit kunt laten zien’ als rationele actoren door het informa-aspect en het forma-aspect van informatie eenduidig aan elkaar te koppelen (hetgeen met behulp van programmeertalen gebeurt). Daarmee werd de weg gebaad voor de ontwikkeling van ‘echte’ informatiesystemen. En thans zitten we dus aan het begin van de derde evolutiegolf, het ‘ontdekken’ en ‘toe-eigenen’ van het B-systeem van een organisatie. Naar mijn mening is het feit dat we nu in deze evolutiefase zitten een complicerende factor in de huidige discussie over architectuur in de ICT. Tegelijkertijd is het mede de aanleiding voor die discussie en zullen we dus moeten leven met deze extra portie complexiteit.

Cruciaal voor het goed laten verlopen van de huidige evolutiefase is het onderkennen dat een B-systeem van een andere categorie is dan een I-systeem, net zoals het in de jaren zeventig cruciaal was te onderkennen dat een I-systeem van een andere categorie is dan een D-systeem. Wie die tijd heeft meegemaakt, weet dat zoiets verre van eenvoudig, verre van triviaal is. Het heeft zeker tien tot vijftien jaar geduurd voordat conceptuele schema’s van databases werkelijk conceptueel waren en voordat logische stroomschema’s het predicaat logisch echt verdienden. Het was, en is, gewoon lastig om zuiver conceptueel te denken over informatieopslag, om niet de afbeelding ervan (in bits en bytes en velden en records) er ook bij te betrekken. Het was, en is, gewoon lastig zuiver logisch te denken over informatietransport, om niet de informatiedragers (de pakketjes, de formulieren enz.) er ook bij te betrekken. Net zo lastig is het nu om B-systemen te begrijpen op een eigen manier, d.w.z. los van de wijze waarop ze worden ondersteund door I-systemen, onafhankelijk dus van informatiestromen en onafhankelijk van het bewaren en berekenen van informatie. Dat vraagt om kennis die nog grotendeels moet worden ontwikkeld, om kennis die in de bestaande opleidingen derhalve nog nauwelijks wordt overgedragen. Dat het wel kan, wordt onder andere bewezen door de groeiende appreciatie van de DEMO-methodiek, waarvan deze wijze van denken de grondslag vormt.

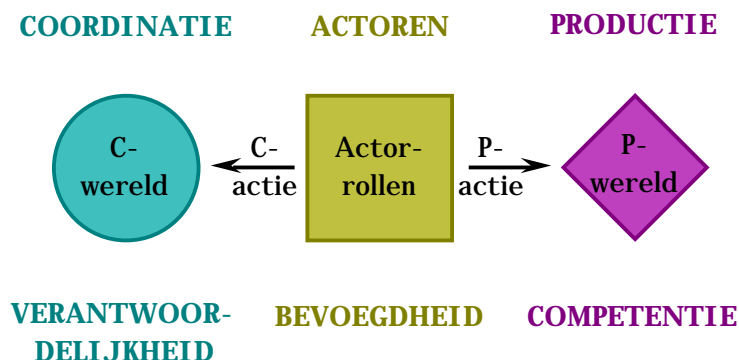
De rechterkant van figuur 1 laat de toevoeging van de functie-oriëntatie en de constructie-oriëntatie aan de drie aspectsystemen zien. Het maakt tevens de samenhang tussen de drie aspectsystemen inzichtelijk. De overvloeiing van de kleuren in de ‘functie-blokjes’ drukt uit dat hier twee systemen (van twee verschillende categorieën) bij elkaar gebracht worden. De interpretatie van de rood-groene overvloeiing in het functie-blokje van het I-systeem is, dat de actoren in het I-systeem (bijvoorbeeld de boekhouders) informatiediensten leveren aan de actoren in het B-systeem (bijvoorbeeld de inkopers), en omgekeerd, dat de actoren in het B-systeem informatiediensten vragen van de actoren in het I-systeem. De interpretatie van de groen-blauwe overvloeiing in het functie-blokje van het D-systeem is, dat de actoren in het D-systeem (bijvoorbeeld de interne postbodes) documentele diensten leveren aan de actoren in

het I-systeem (bijvoorbeeld de boekhouders), en omgekeerd, dat de actoren in het I-systeem documentele diensten vragen van de actoren in het D-systeem. De interpretatie van de witrode overvloeiing in het functie-blokje van het B-systeem is, dat de vervulling van de functie van het B-systeem in overeenstemming is met het doel of de missie van de organisatie.

De kerntaak van de functie-ontwerper bestaat uit het optimaal laten passen van het systeem ‘aan de onderkant’ bij het systeem ‘aan de bovenkant’. Het ontwerpen van de functie van een systeem is dus in de meest letterlijke zin het ontwerpen van een interface, van de twee zijden van een medaille, maar wel een interface tussen twee systemen van een verschillende categorie! Bijvoorbeeld, het ontwerpen van de functie van een D-systeem vraagt om grondige kennis van de constructie van D-systemen en om grondige kennis van de constructie van I-systemen. De kennis van de constructie van het I-systeem is de basis voor het vaststellen van de requirements voor het te ontwerpen D-systeem. Met het ontwerpen van de functie van het D-systeem beoogt de ontwerper te voldoen aan de requirements. Een soortgelijke redenering gaat op voor de relatie tussen het I-systeem en het B-systeem.

4 Een universele constructie-architectuur voor organisaties

Het is mogelijk een universele, dus altijd en overal toe te passen, architectuur op te stellen voor organisaties. De hier gepresenteerde architectuur is ontleend aan de DEMO methodiek; zij is geworteld in de systeemontologie van Mario Bunge en de communicatieve actietheorie van Jürgen Habermas. Figuur 2 toont een schematische voorstelling van de architectuur.



Figuur 2 De universele constructie-architectuur voor organisaties

De systeemcategorie waartoe organisaties behoren is die van de sociale systemen, d.w.z. van systemen waarvan de elementen sociale individuen ofwel subjecten zijn (mensen dus in hun hoedanigheid van sociaal wezen). Deze subjecten verrichten twee soorten acties: productie-acties (afgekort tot P-acties) en coördinatie-acties (afgekort tot C-acties). Door het verrichten van *productie-acties*, dragen de subjecten bij aan het voortbrengen van de goederen of diensten die de organisatie aan haar omgeving levert, en waardoor dus uiteindelijk de functie of missie van de organisatie wordt waargemaakt. Een productie-actie kan materieel van aard zijn of immaterieel. Voorbeelden van materiële acties zijn alle soorten fabricage-handelingen alsmede handelingen ten behoeve van de opslag en het transport van goederen. Voorbeelden van immateriële acties zijn de veroordeling van een verdachte door een rechtbank, de beslissing een claim op een verzekering goed te keuren, en de benoeming van een burgemeester. Door het verrichten van *coördinatie-acties*, gaan subjecten *commitments* (afspraken, overeenkomsten) jegens elkaar aan en komen ze die na. Die commitments betreffen altijd de verrichting van productie-acties. Het begrip coördinatie dat hier wordt

gebruikt is heel breed. Het omvat niet alleen de onderlinge afstemming van werk tussen de medewerkers in een bedrijf, maar ook de interactie tussen hen en de klanten en leveranciers van het bedrijf, de subjecten in de omgeving dus. Die interacties verschillen namelijk niet wezenlijk, alleen in de wijze van realisatie.

Het resultaat van het met succes uitvoeren van een productie-actie is een *productiefait* ofwel P-feit. Voorbeelden van productiefaiten (in de context van een verzekeringsbedrijf) zijn 'polis P is afgesloten' en 'polis P is betaald'. De variabele P staat daarin voor een bepaalde instantiatie van het entiteitstype polis, voor een bepaalde verzekering dus. Om op een verantwoorde manier een nieuw productie-feit te creëren moet in het algemeen aan een aantal voorwaarden zijn voldaan. Die voorwaarden zijn altijd uit te drukken in het bestaan van andere productiefaiten. In het gegeven voorbeeld zal waarschijnlijk het bestaan van het feit 'polis P is betaald' een voorwaarde zijn voor het creëren van het feit 'polis P is afgesloten'. Met andere woorden, er is altijd een bepaalde *productstructuur*, niet alleen voor materiële productie maar ook voor immateriële.

Om te kunnen abstraheren van het bepaalde subject dat een actie verricht en daarmee zich te kunnen concentreren op de organisatorische rol die het subject door het verrichten van die actie vervult, voeren we het begrip actorrol in. Een *actorrol* is de 'portie' bevoegdheid om een bepaald soort productie-acties te verrichten, bijvoorbeeld het afsluiten van polissen. In het algemeen vallen actorrollen in een organisatie niet samen met de bestaande functies in de organisatie. Die zijn nu eenmaal op een andere manier tot stand gekomen. Het is wel altijd mogelijk actorrollen op organisatiefuncties af te beelden. Een actorrol kan door meerdere subjecten worden vervuld en een subject kan meerdere actorrollen vervullen. Een subject in de vervulling van een bepaalde actorrol heet een *actor*.

In overeenstemming met het gemaakte onderscheid tussen productie-acties en coördinatie-acties, onderscheiden we twee werelden (universes of discourse) waarin de twee soorten acties effect hebben: de *productiewereld* ofwel P-wereld en de *coördinatie wereld* ofwel C-wereld (vergelijk figuur 2). Beide werelden bevinden zich op elk moment in een bepaalde toestand. Zo'n toestand is gedefinieerd als de verzameling (P- dan wel C-) feiten die op dat moment het geval zijn. Het creëren van een nieuw feit, bijvoorbeeld het afsluiten van een polis, is een toestandverandering ofwel *transitie*, in dit geval een transitie in de P-wereld van het verzekeringsbedrijf. Een bepaalde transitie op een bepaald moment heet een *event* (N.B. een bepaalde transitie kan dus in principe, in de loop der tijd, meerdere keren voorkomen).

Actoren zijn de 'motors' van een organisatie. Zij zijn intrinsiek alert op wat ze kunnen doen en ze proberen voortdurend hun *agenda* af te werken (hun werkvoorraad). Elk agendum (enkelvoud van agenda) is een C-feit waaraan de actor zich heeft gecommitteerd. Als iemand bijvoorbeeld aan een verzekeringsagent vraagt een verzekering af te sluiten, dan is dat verzocht zijn een agendum voor de verzekeringsagent. Bij elk soort agendum hoort een bepaalde actieregel die een richtlijn is voor de uitvoerende actor. Als uitkomst van het afwerken van het agendum 'het afsluiten van polis P is verzocht', kan de verzekeringsagent beloven de verzekering af te sluiten of dat afwijzen. Het resulterende C-feit (beloofd of afgewezen) is weer een agendum voor een actor. In de actieregel kan staan dat er alvorens te kunnen beloven eerst andere C-acties moeten worden verricht, bijvoorbeeld het verzoek (aan een andere actor) om vast te stellen dat de aanvrager gerechtigd is. Op die manier kan er dus een cascade van C-acties en P-acties ontstaan. Zo'n cascade heet een *bedrijfsproces*. De structuur daarvan weerspiegelt de productstructuur van de geleverde dienst.

Onderaan in figuur 2 worden de begrippen verantwoordelijkheid, bevoegdheid en competentie genoemd. Bevoegdheid is al aan de orde gekomen bij het definiëren van het

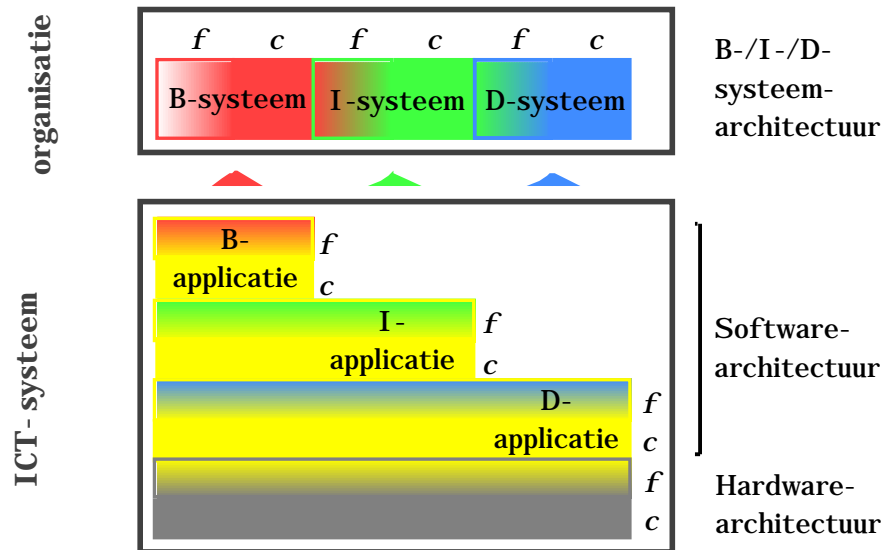
begrip actorrol. We zullen nu alle drie de begrippen wat preciezer beschrijven alsook de onderlinge samenhang. Met *competentie* wordt bedoeld de kunde (vaardigheid, kennis enz.) om een bepaald soort productie-acties uit te voeren. Bijvoorbeeld, als je loodgieter of arts of rechter bent, ben je in staat bepaalde P-acties te verrichten. Je hebt op de een of andere manier de benodigde kennis en ervaring opgedaan om dat te kunnen. Competentie is iets wat je 'onvervreemdbaar' bezit, ook al doe je ermisschien niets mee. Om je beroep of vak te kunnen uitoefenen moet je in dienst treden van een institutie (een bedrijf, een stichting, de overheid enz.). Doordat je in dienst wordt genomen van een organisatie, verkrijgt je de *bevoegdheid* namens die organisatie je vak te beoefenen. Zo kun je bijvoorbeeld loodgieter zijn bij bedrijf X of arts bij ziekenhuis Y of rechter bij rechtbank Z. Krachtens de cultuur (de waarden en normen) van de organisatie, alsook krachtens de maatschappelijke waarden en normen, wordt van je verwacht dat je de toegekende bevoegdheid op een *verantwoorde* wijze gebruikt. De loodgieter, bijvoorbeeld, wordt geacht op een verantwoorde wijze de klanten van bedrijf X tegemoet te treden. Hetzelfde geldt voor de arts jegens de patiënten van ziekenhuis Y en voor de rechter jegens de aan rechtbank Z voorgeleide verdachten.

Met de keuze van de drie primaire kleuren (cyaan, magenta en geel) in figuur 2, bedoel ik twee dingen uit te drukken. Het ene is dat de drie componenten (actoren, coördinatie en productie) een complete set van concepten vormen om de constructie en de werking van organisaties te begrijpen; ze zijn dus alledrie noodzakelijk en tevens voldoende. Het andere is, dat het model van figuur 2 orthogonaal is aan dat van figuur 1 (vandaar de keuze voor een andere basisset van primaire kleuren). Dat betekent dus dat de organisatie-architectuur van toepassing is op elk van de drie aspectsystemen. Gezien vanuit de gepresenteerde constructie-architectuur van organisaties zit hem het verschil tussen de aspectsystemen in de aard van de productiefeiten en vervolgens ook in de aard van de actorrollen (ofwel in de aard van de vereiste competenties en vervolgens in de aard van de bevoegdheden). De overeenkomst tussen de drie aspectsystemen zit aan de coördinatiekant. In alle drie de systemen zijn subjecten de actieve elementen en berust de werking van de systemen dus op het 'spel' van het aangaan en nakomen van commitments.

De verbindende pen (linking pin) tussen de aspectsystemen wordt gevormd door het feit dat een subject tegelijkertijd in twee of drie aspectsystemen coördinatie-acties kan verrichten. Ik zal dit illustreren aan de hand van het eerder gebruikte scenario van het verzekeringsbedrijf, nu even heel concreet gemaakt door een bepaalde persoon (Piet) als cliënt te nemen en een bepaalde persoon (Marie) als verzekeringsagent. Piet en Marie zitten tegenover elkaar in een spreekkamer. Piet vraagt Marie een nieuwe verzekering af te sluiten. Marie is bevoegd de rol van verzekeringsafsluiter te spelen doordat het bedrijf haar deze bevoegdheid heeft verleend. Piet is bevoegd tot het spelen van de rol van cliënt op grond van een algemene maatschappelijke 'bevoegdheidsverlening'. Beide rollen zijn 'rode' rollen, d.w.z. actorrollen in het B-systeem van het verzekeringsbedrijf. Bij het afwerken van het agendum dat door het verzoek van Piet is ontstaan, kan Marie het nodig vinden dat ze Piet's geboortedatum kent. Dat moet ze dan vragen aan de creator of eigenaar van dat feit. Daarmee belanden we in het I-systeem. Marie, als ondeelbaar subject, hult zich nu in haar 'groene' gedaante en vraagt aan een andere I-actor om de geboortedatum van Piet. Waarschijnlijk is die andere actor Piet zelf (nu in zijn 'groene' gedaante). Merk evenwel op dat Piet niet de creator of eigenaar van zijn geboortedatum is. Dat is namelijk de ambtenaar van de burgerlijke stand in zijn geboorteplaats. Het zou dus beter zijn dat Marie bij die ambtenaar navraag zou doen (al of niet via het middel van het GBA-systeem). Tijdens het gesprek van Piet en Marie vult Marie een formulier in. Marie toont dan haar 'blauwe' gedaante, ze verricht namelijk een documentele actie. Na afloop van het gesprek stopt ze het ingevulde formulier in een envelop en geeft die aan Karel, de interne postbode. Karel is een zuivere 'blauwe' actor, d.w.z. hij vervult in het verzekeringsbedrijf geen rollen in het I-systeem of het B-systeem.

6 Een raamwerk voor ICT-architectuur

In het voorgaande zijn de drie aspectsystemen van een organisatie geheel los van hun realisatie ofwel inrichting besproken, d.w.z. technologie-onafhankelijk. Een interessante vraag is natuurlijk wat er aan figuur 2 moet worden veranderd of uitgebreid om met name de inbreng van ICT in de inrichting uit te beelden. Dat is in figuur 3 geprobeerd. In het bovenste deel (organisatie) is het model van figuur 2 overgenomen, alleen nu horizontaal in plaats van verticaal. De 'f' staat voor de functie-oriëntatie en de 'c' voor de constructie-oriëntatie. Het resultaat is een *raamwerk* voor ICT-architectuur. Vergelijkbare raamwerken zijn dat van John Zachman en het Integrated Architecture Framework van Hans Goedvolk e.a. Een recente interessante studie is die van Jan Hoogervorst, die op het LAC 2002 wordt gepresenteerd.



Figuur 3 Een raamwerk voor ICT-architectuur

ICT-systemen zijn geen componenten van een B-systeem, een I-systeem, of een D-systeem zoals figuur 3 duidelijk laat zien. Ze kunnen wel de actoren in die systemen ondersteunen. Dit is een voor de praktijk van de ICT belangrijke constatering. Het impliceert namelijk dat de verantwoordelijkheid van de B-, I- en D-actoren nooit kan worden gedelegeerd aan ICT-systemen. Hoe vergaand de ondersteuning van de interne postbode (let wel: de actorrol, niet de persoon) ook is, bijvoorbeeld doordat het verzekeringsbedrijf papierloos is geworden, de verantwoordelijkheid voor het correct functioneren van het e-mailsysteem blijft bij de interne postbode berusten. We noemen die rol misschien niet meer interne postbode, maar bijvoorbeeld 'manager interne communicatie', de essentie is daar echter niet door veranderd. Misschien is het beheer van het interne e-mailsysteem compleet uitbesteed, ook dan echter behoudt de interne postbode of de manager interne communicatie zijn verantwoordelijkheid. Omdat zijn directe invloed op het functioneren van het systeem in die situatie gering is, zal hij voor een zodanig contract moeten zorgen dat hij een deel van zijn verantwoordelijkheid kan 'verhalen' bij de externe beheerder. Het kan ook zijn dat het e-mailsysteem helemaal intern wordt beheerd, door de eigen ICT-afdeling, bijvoorbeeld omdat het daar is gebouwd. Die situatie lijkt heel anders te zijn dan de vorige, maar dat is schijn. Ook nu moet er een contract over de dienstverlening worden gesloten. Kortom, functioneel beheer behoort niet alleen het beheer van de functionaliteit te zijn maar ook dat van het operationeel functioneren.

Vervolgens, hoe vergaand de ondersteuning van de boekhouder (de actorrol dus) ook is, bijvoorbeeld doordat iedereen werkt met een boekhoudapplicatie, de verantwoordelijkheid voor het correct functioneren van de boekhoudapplicatie blijft onveranderd bij de boekhouder berusten. Wat er voor de boekhouder, net zo als voor de interne postbode, wel is veranderd, is dat een heleboel werk door het ICT-systeem wordt gedaan. Dat werk betreft onder andere het rekenen en het opslaan en transporteren van data. En dat is prima, ICT-systemen zijn daarin nu eenmaal beter dan mensen. Alleen van de actoren in het B-systeem kunnen ICT-systemen geen werk overnemen. Men kan een ICT-systeem nu eenmaal niet laten verkopen of inkopen of welk ander soort originele feiten ook laten creëren. Er is wel vergaande ondersteuning mogelijk. Een voorraadbeheersysteem kan bijvoorbeeld perfect de optimale bestelhoeveelheid voor een artikel uitrekenen, maar het is de beslissing van de inkoper die hoeveelheid ook te bestellen. Dat is zijn verantwoordelijkheid, en die is wezenlijk verschillend van die van de boekhouder die de omzet van de afgelopen periode berekent. Het getal dat de boekhoudapplicatie produceert is de omzet, want omzet is een welgedefinieerd afgeleid gegeven. Men kan toch moeilijk de boekhouder ervoor verantwoordelijk stellen dat $1+1=2$.

In figuur 3 zijn binnen het blok 'ICT-systeem' drie soorten applicaties onderscheiden: B-applicaties, I-applicaties en D-applicaties. Aan elk is vervolgens de functie- en de constructie-oriëntatie toegevoegd. Met de constructie-oriëntatie op de software van een ICT-systeem, die steeds geel is gekleurd, wordt bedoeld het concrete (computer)programma, uitgedrukt in een formele taal. Die taal mag variëren van machinecode tot SQL. In de hardware van een ICT-systeem, die volledigheidshalve aan het raamwerk is toegevoegd, krijgt dat programma zijn operationele verwerking (wordt het geëxecuteerd). Daarom is de functie-oriëntatie op de hardware als een overvloeiing van geel naar grijs gekleurd (grijs is de gekozen kleur voor hardware). De betekenis van de kleurovervloeiingen in de functie-oriëntaties van de B-, de I- en de D-applicatie is analoog aan die van de kleurovervloeiingen in het B, het I- en het D-systeem.

Een D-applicatie ondersteunt het werk van D-actoren. Typische voorbeelden van D-applicaties zijn tekstverwerkers, e-mailsystemen en DBMS-en. Het kenmerk van D-systemen is er meteen in te herkennen: het gaat om iets doen met documenten, met informatie, maar dan zonder je om de inhoud ervan te hoeven bekommeren. Typische voorbeelden van I-applicaties zijn boekhoudapplicaties, statistiekpakketten en alle managementinformatiesystemen. Dat zijn allemaal 'zuivere' informatiesystemen, d.w.z. ICT-applicaties die de toestand van (een deel van) het B-systeem van een organisatie registreren en daarover informatie verstrekken, inclusief alle afgeleide informatie. Zij ondersteunen actoren in het I-systeem. B-applicaties tenslotte zijn allemaal beslissingsondersteunende systemen, zoals budgetteringssystemen, verkoopsystemen en procesmanagementsystemen. Hoewel vaak de schijn wordt gewekt dat de B-applicaties zelf beslissingen nemen, kan dat per definitie niet het geval zijn, zoals hiervoor is aangetoond. Een inkoopbestelling, bijvoorbeeld, die door een automatisch bestelsysteem wordt geplaatst, vat de beoogde leverancier heus op als een echte bestelling, d.w.z. als iets waarvoor een echte actor zich heeft geëngageerd, ofwel 'waarachter' een echte inkoper zit.

De 'balk' van de D-applicaties loopt naar links door. Daarmee wordt uitgedrukt dat sommige D-applicaties ook worden gebruikt in of misschien zelfs speciaal zijn bedoeld voor de implementatielaag van I-applicaties. DBMS-en en netwerksystemen zijn daar typische voorbeelden van. Met het doorlopen van de 'balk' van de I-applicaties wordt uitgedrukt dat sommige I-applicaties (ook) worden gebruikt als implementatielaag voor B-applicaties. In principe, kan men stellen, dat elke B-applicatie, bijvoorbeeld een ERP-pakket, is gebouwd op een registratiesysteem, en dat elk registratiesysteem is gebouwd op een systeem voor het opslaan en transporteren van informatie.

Een actuele tendens in de ICT is het integreren van de D-applicaties in een bedrijf. Samen met de hardware worden ze gewoonlijk de *ICT-infrastructuur* genoemd. Het liefst moet die infrastructuur compatibel zijn met die van andere bedrijven. Om dat te bereiken is het nodig dat D-applicaties zonder problemen aan elkaar gekoppeld kunnen worden. Hier blijkt het grote belang van functie-architectuur, van standaards dus en van het zich conformeren aan standaards (normaliseren). Gelukkig is iedereen het daar wel over eens, dat er standaarden moeten komen voor de vorm (de structuur) van documenten, dat er dus een, liefst mondiale, *functie-architectuur* voor infrastructuur moet komen. Het is onmiskenbaar dat XML hierbij hoge ogen gooit. Maar dat is natuurlijk niet voldoende. De volgende stap is een functie-architectuur voor I-applicaties en vervolgens waarschijnlijk het ontstaan van een informatie-infrastructuur. Een pionier is er al: het Semantisch Web. Uiteindelijk willen bedrijven op een betrouwbare wijze, en liefst ook op een mondiale schaal, met elkaar elektronisch zaken doen. Ook al wil de hype rondom e-business ons doen geloven dat we er al zijn, de werkelijkheid is dat we er nog lang niet zijn. Daarvoor is namelijk nóg een standaard nodig, een standaard voor de functie-architectuur van B-applicaties, en die is er nog niet. Het raamwerk in figuur 3 laat wel de richting zien waar men die standaard moet zoeken. Die is dat hij gebaseerd moet zijn op (of in overeenstemming moet zijn met) de patronen van coördinatie in een B-systeem.

Geraadpleegde bronnen en achtergrondliteratuur

- Britton, C., *IT architectures and middleware: strategies for building large, integrated systems*, Addison-Wesley, 2001
- Bunge, M.A., *Treatise on Basic Philosophy*, vol.3, *The Furniture of the World*, D. Reidel Publishing Company, Dordrecht, The Netherlands, 1977
- Bunge, M.A., *Treatise on Basic Philosophy*, vol.4, *A World of Systems*, D. Reidel Publishing Company, Dordrecht, The Netherlands, 1979
- Dietz, J.L.G. De skills van de ICT-architect, *Proceedings van LAC 2001*, <zie de website van LAC 2001: www.serc.nl/lac/LAC-2001>
- DEMO methodiek <zie de website www.demo.nl>
- Goedvolk, J.G., H. de Bruin and D.B.B. Rijsenbrij, Integrated Architectural Design of Business and Information Systems, *Proceedings of the Second Nordic Workshop on Software Architecture (NOSA'99)*, 1999
- Habermas, J., *Theorie des Kommunikativen Handelns*, Erster Band, Suhrkamp Verlag, Frankfurt am Main, 1981
- Hoogervorst, J.A.P., Managing Enterprise Architecture bij KLM, <zie de LAC 2002 website: www.serc.nl/lac>
- Rees, J. van, J. Schekkerman, Het alternatief LAC <zie de LAC 2002 website: www.serc.nl/lac>
- Röling, W., *De kunst van de ingenieur*, Salomé - Amsterdam University Press, 2002
- Searle, J.R., *The Construction of Social Reality*, Allen Lane, The Penguin Press, London, 1995
- Stamper, R., *Information in Business and Administrative Systems*, B.T. Batsford, London, 1973
- Zachman, J.A., A framework for information systems architecture. *IBM Systems Journal*, 26(3):276-292, 1987.