

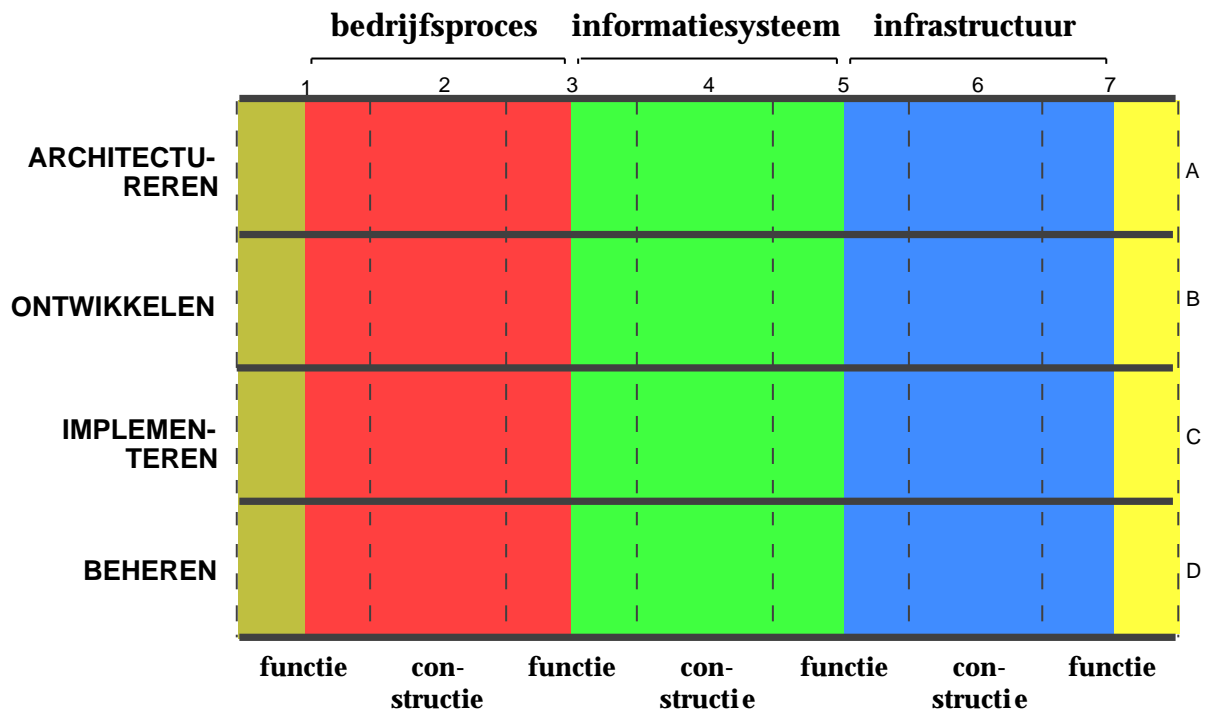
# De informatie-architect op zijn plaats gezet

## 1 Inleiding en verantwoording

In juni 1996 heeft de Verkenningcommissie Informatica (VIN) het rapport "Geen toekomst zonder informatica" uitgebracht [1]. Daarin staat het onderwerp architectuur opgenomen in de 'toplijst' van vijftien aandachtspunten voor onderzoek, met de volgende nadere omschrijving: "Er is grote behoefte aan methoden om op een hoog abstractieniveau de inrichting van de informatievoorziening van een bedrijf of de inrichting van de besturing van een apparaat te kunnen beschrijven en erover te kunnen redeneren. Dit is het werk van systeemarchitecten. Deze beschrijvingen noemen we architecturen." In februari 1998 is het rapport "De structurele behoefte aan informatici" van de Adviesraad voor het Wetenschaps- en Technologiebeleid" [2] verschenen. Daarin wordt een grote behoefte gesignaleerd aan architectenfuncties. In het pleidooi voor architectenopleidingen dat daarin wordt gehouden, staat onder andere: "De verwevenheid van informatietechnologie met allerlei bedrijfsprocessen en de complexiteit van informatiesystemen vragen om aparte deskundigen die in staat zijn informatiesystemen te ontwerpen die een passend antwoord zijn op de eisen van de gebruiker." In de motivering wordt verwezen naar het eerder genoemde VIN-rapport, en naar de in oktober 1997 verschenen "Nationale Onderzoeksagenda Informatica" van de Stichting Informatica Onderzoek Nederland (SION) [3], in het bijzonder naar het daarin genoemde onderzoeksthema 'Requirements Engineering', dat als volgt wordt gedefinieerd: "Activiteiten die tot doel hebben de eisen en wensen van gebruikers en andere betrokkenen van een informatieverwerkend systeem te achterhalen en te specificeren als onderdeel van de ontwikkeling c.q. aanpassing van dat systeem". In datzelfde rapport wordt overigens ook software-architectuur als thema besproken, met de volgende definitie: "De software architectuur van een informatiesysteem is de verdeling van taken, eigenschappen en gedragsregels over de verschillende bouwstenen en communicatie-mechanismen die tezamen de gewenste functionaliteit van het systeem realiseren".

Over het belang van het beroep informatie-architect en over het belang van meer aandacht voor architectuur in het algemeen, lijken de gezaghebbende partijen het dus wel eens te zijn. Daarover hoeven de oprichters van het Nederlands Genootschap voor Informatie-Architecten (NGIA) zich geen zorgen meer te maken. Maar wat is een informatie-architect eigenlijk, en bedoelt iedereen die erover spreekt wel hetzelfde? Dat is de vraag waarop ik in dit artikel probeer een antwoord te geven. Dat antwoord zal en kan niet absoluut zijn, het is hopelijk wel een zinvolle bijdrage aan de collectieve meningsvorming. Uiteindelijk is een informatie-architect namelijk wat informatie-architecten vinden dat het is! Ik zal het begrip architectuur nader omschrijven, en ik zal het beroep van informatie-architect plaatsen in relatie tot andere (deel)beroepen binnen de informatica. Daartoe maak ik gebruik van een generieke structuur van het werkkterrein Informatica die in de afgelopen jaren, mede op grond van enkele eerdere studies binnen het NGI (Nederlands Genootschap voor Informatica) en de VRI (Vereniging van Register-Informatici), is ontwikkeld, en die

ik L\_PASO<sup>1</sup> heb gedoopt. Een voorloper van L\_PASO is gepubliceerd als het VRI-model [4]. Ter validering van het raamwerk zijn onder andere ISM<sup>2</sup>, EISS<sup>3</sup> en TFBI<sup>4</sup> in de structuur van L\_PASO geplaatst [5]. L\_PASO is inmiddels ook door het NGI geadopteerd, en het kan zich verheugen in de belangstelling van CEPIS (Council of European Professional Informatics Societies) en van COCKPIT (Centraal Overleg Certificatie van Kwaliteit van Professionals in de IT). In L\_PASO wordt het werkkerrein Informatica gestructureerd in drie dimensies: soorten systemen, systeem-oriëntaties en soorten werkzaamheid. Die worden respectievelijk in paragraaf 2, 3 en 4 besproken, aan de hand van figuur 1.



Figuur 1 De structuur van het werkkerrein Informatica

In paragraaf 5 wordt nader ingegaan op hoe je de structuur van figuur 1 kunt gebruiken om deelberoepen of werkdomeinen binnen de Informatica te definiëren, waarna dat in paragraaf 6 wordt gedaan voor het beroep informatie-architect.

<sup>1</sup> L\_PASO omvat meer, het is namelijk ook een systematiek voor de classificatie van functies. Voor dit artikel is echter alleen de structuur van het werkkerrein van belang.

<sup>2</sup> "Industry Structure Model", opgesteld en beheerd door de British Computer Society. Inmiddels is de derde versie verschenen (ISM/3).

<sup>3</sup> "European Informatics Skills Structure", een variant op ISM/2.

<sup>4</sup> "Taken en Functies in de Bestuurlijke Informatica", geproduceerd en uitgegeven door het NGI in 1986.

## 2 Soorten systemen

### *Infrastructuur*

Het eerste en oudste deelgebied binnen de Informatica betreft de opslag, het transport en de verwerking van data. Door de 'uitvinding' van de binaire code, en door de ontwikkeling van daarop gebaseerde coderingsstelsels (zoals ASCII en EBCDIC), was het mogelijk geworden elektrische, magnetische en optische media aan te wenden voor de opslag van ongekend grote hoeveelheden data, en voor het transport ervan met ongekend grote snelheden. Dezelfde 'uitvinding' maakte het ook mogelijk dat data met behulp van digitale elektronische apparaten zeer efficiënt kon worden verwerkt (algoritmische berekeningen en logische afleidingen). De inspanningen van degenen die zich op dit aandachtsgebied hebben toegelegd, hebben geresulteerd in zeer geavanceerde en breed toepasbare software-systemen. Voorbeelden van zulke systemen zijn DBMS-en, e-mailsystemen en programmatheken (inclusief de moderne variant van object libraries). Deze systemen vormen de bouwstenen van wat tegenwoordig de (ICT-) infrastructuur wordt genoemd. In figuur 1 beslaan ze het blauwe vak. Het belangrijke gezamenlijke kenmerk van de 'blauwe' systemen, is dat ze *toepassings-onafhankelijk* zijn. Anders gezegd, de betekenis van de opgeslagen, getransporteerde of geproduceerde informatie voor een daarmee ondersteund applicatiedomein speelt geen rol.

### *Informatiesystemen*

Dat is juist wel het geval in het tweede aandachtsgebied. Gebruik makend van de infrastructurale componenten, legt men zich in dit aandachtsgebied toe op de ontwikkeling van informatiesystemen ofwel applicaties. Zij worden in figuur 1 voorgesteld door het groene vak. Informatiesystemen zijn steeds specifiek voor een bepaald applicatiedomein. Centraal in dit deelgebied van de Informatica staat de gedachte dat een informatiesysteem een afbeelding bevat van dat applicatiedomein. Aan de waarheids-getrouwheid (waaronder zijn begrepen zaken als relevantie, precisie en actualiteit) van deze afbeelding worden informatiesysteem-specifieke eisen gesteld. Die eisen worden afgeleid van de kenmerken van de (bedrijfs)activiteiten waarbij de informatie wordt gebruikt. Ter illustratie: de eisen die aan een informatiesysteem voor de luchtverkeersbegeleiding worden gesteld zullen veel strenger zijn dan de eisen die men stelt aan een systeem dat de ledenadministratie van de tennisclub ondersteunt.

### *Bedrijfsprocessen*

Het derde en jongste aandachtsgebied van de Informatica wordt gevormd door de uitvoerende en bestuurlijke activiteiten in een organisatie. De gedachte die nu centraal staat, is dat een informatiesysteem onderdeel is van de inrichting van een bedrijfsproces, het ondersteunt namelijk de eigenlijke uitvoerder(s) daarvan. De ontwerper van een informatiesysteem dient daarom een gedegen inzicht te hebben in het ondersteunde bedrijfsproces. Daarnaast is het zo, dat de effectiviteit van de bijdrage van een informatiesysteem aan het functioneren van een organisatie, mede afhangt van de kwaliteit van het ondersteunde bedrijfsproces zelf. In figuur 1 beslaan bedrijfsprocessen het rode vak.

Lange tijd zijn bedrijfsprocessen alleen bestudeerd binnen de organisatiekunde, de bedrijfskunde en de bedrijfseconomie, ze kunnen echter ook vanuit een

informatiekundig perspectief worden bestudeerd en begrepen. Een bedrijfsproces is dan een aaneenschakeling van afspraken tussen actoren, en die afspraken zijn het resultaat van communicatieve acties zoals verzoeken (opdrachten, orders, bestellingen etc.), beloftes (offertes, orderacceptaties, toezeggingen etc.), en verklaringen en aanvaardingen van resultaten van acties.

De bijdrage die informatici op deze wijze kunnen leveren aan het (her)ontwerpen en (her)inrichten van bedrijfsprocessen, is complementair aan de organisatiekundige bijdrage, die is gebaseerd op organisatiekundige of bedrijfseconomische theorieën (zoals de economische waarde-keten, succes/prestatie-indicatoren en logistieke principes).

#### *Grensgebieden*

Er bestaan twee belangrijke grensgebieden van de Informatica met andere vakgebieden. Het ene grensgebied bestaat uit het op elkaar afstemmen van de missie van een organisatie en de bedrijfsprocessen die voor de vervulling van de missie zorgen. In dit grensgebied ontmoeten Informatici de al eerder genoemde Bedrijfskundigen, Bedrijfseconomen en Organisatie-kundigen (kolom 1 in figuur 1). Het andere grensgebied bestaat uit het op elkaar afstemmen van het formele (syntactische) aspect van informatie en de fysische substanties waarin de vorm van informatie zijn neerslag vindt. In dit grensgebied vindt de ontmoeting plaats van Informatici met Electrotechnici (vooral de Electronici) en Mechanici, met de hardware-mensen dus (kolom 7 in figuur 1).

### **3 Systeem-oriëntaties**

In het bestuderen van een systeem kan men twee, fundamenteel verschillende, oriëntaties innemen: de functie-oriëntatie en de constructie-oriëntatie. De kennis die men in elk van deze oriëntaties verkrijgt is complementair. Om het onderscheid tussen beide uit te leggen, nemen we de auto als voorbeeld en kijken we naar de verschillende soorten kennis die een chauffeur en een monteur over auto's hebben.

#### *Functie-oriëntatie*

Een chauffeur is iemand die een auto bestuurt, erin rijdt. Dat kan hij op grond van de functionele kennis die hij bezit van de auto. Die kennis is geworteld in een *black-box-model* van de auto. In zo'n model wordt de invloed van de omgeving op het systeem voorgesteld door (waarden van) invoervariabelen, en de response daarop door (waarden van) uitvoervariabelen. Het verloop van de waarden van deze variabelen in de tijd heet het gedrag van het model. Vandaar dat het verband tussen de invoeren de uitvoervariabelen de gedragsfunctie wordt genoemd. Hoe dat gedrag tot stand komt is vanuit de functie-oriëntatie gezien irrelevant. De chauffeur kan volstaan met te weten welke effecten veranderingen in de waarden van de invoervariabelen hebben op de waarden van de uitvoervariabelen. Voorbeelden van de invoervariabelen zijn de stand van het gaspedaal en de stand van het stuurwiel. Voorbeelden van de uitvoervariabelen zijn de snelheid van de auto en de richting waarin hij beweegt. Het hanteren van een black-box-model komt overeen met het innemen van de *functie-oriëntatie*.

### *Constructie-oriëntatie*

Een monteur repareert een auto en onderhoudt hem. Dat doet hij op grond van de constructionele kennis die hij van de auto heeft. Die kennis is geworteld in een *white-box-model* van de auto. Het komt er op neer dat hij precies weet uit welke componenten de auto is opgebouwd, wat de *werking* is van elke component, en hoe de werkingen van de componenten gezamenlijk ervoor zorgen dat de functie van de auto wordt waargemaakt. Die componenten kent de monteur in principe slechts vanuit de functie-oriëntatie. Zo is het voldoende dat hij de functie/het gedrag kent van accu's, O-ringen en V-snaren. Hij hoeft ze niet zelf te kunnen maken. Anders gezegd, op eenzelfde wijze als de coureur gebruiker is van de auto, is de monteur gebruiker van deze componenten. Het hanteren van een *white-box-model* komt overeen met het innemen van de *constructie-oriëntatie*.

Wat voor auto's geldt, geldt ook voor de eerder besproken soorten systemen, bedrijfsprocessen, informatiesystemen en infrastructurele componenten. Om welk systeem dan ook te bestuderen, moet men er altijd eerst een model van hebben of maken, en het soort model bepaalt welke kennis men zich over het systeem verwerft. Tussen de twee extremen van het *black-box-model* en het *white-box-model* zijn diverse 'grijze' modelsoorten mogelijk en gangbaar, waarvan sommige ook binnen de Informatica vrij goed bekend zijn. Voorbeelden daarvan zijn het stroommodel (flow model) en het besturingsmodel (control model).

De twee systeemoriëntaties zijn in figuur 1 voor elk van de drie systeemsoorten opgenomen. De gekleurde systeemvakken zijn met opzet een stukje naar rechts verschoven, om duidelijk te laten zien dat degenen die in een functie-kolom werken, 'boundary spanners' zijn. Het is niet zo dat de geboden functionaliteiten aan 'de rechterkant' vanzelfsprekend passen bij de constructionele behoeften aan 'de linkerkant', en evenmin is het zo dat ze daar op een eenvoudige wijze uit kunnen worden afgeleid. 'Boundary spanning' vereist enerzijds een gedegen constructionele kennis van 'de linkerkant', en anderzijds een voortdurende creativiteit in het vinden en bedenken van de gewenste (en bij de actuele stand van de technologie haalbare) functionaliteit aan 'de rechterkant'.

Daarentegen zou men degenen die in een constructie-kolom werken, 'home keepers' kunnen noemen. Zij hebben als belangrijkste taak ervoor te zorgen dat de functionaliteit van een systeem met de gewenste kwaliteitseisen tot stand komt en in stand blijft.

## **4 Soorten werkzaamheden**

Er worden vier soorten werkzaamheden onderscheiden, te weten architectureren, ontwikkelen, invoeren en beheren. Deze werkzaamheden komt men ook in andere technische disciplines tegen, ze vormen tezamen de bekende 'system life cycle'.

### *Architectureren*

Architectureren bestaat uit het bedenken van de conceptuele kaders waarbinnen degenen die meer concreet bezig zijn met bedrijfsprocessen, applicaties of infrastructurele componenten, dienen te opereren. Architectureren is eerst en vooral het creëren van die architecturen, maar omvat ook het maken van plannen en het uitzetten van strategieën. Een goede vergelijking is die met het werk dat op het gebied van de ruimtelijke ordening gebeurt, zowel op gemeentelijk als op regionaal, provinciaal en landelijk niveau, om te komen tot ontwikkelingsplannen voor de infrastructuur (wegen, spoorlijnen, de distributie van water en energie etc.), voor de stedenbouw (huizen, kantoren, algemene voorzieningen etc.), en voor de verdeling in bebouwde en onbebouwde gebieden. Van een architect wordt verwacht dat hij of zij een brede kennis heeft en ook een brede belangstelling, dat wil zeggen, oog heeft voor de effecten die keuzes en besluiten kunnen hebben buiten het gebied waarop het architectureren direct betrekking heeft. Het werk vereist dat men associatief en strategisch kan denken en creatief is. Daarnaast dient men over een behoorlijke dosis communicatieve vaardigheden te beschikken, ideeën moeten namelijk ook nog verkocht worden.

#### *Ontwikkelen*

Onder ontwikkelen wordt zowel het ontwerpen van systemen verstaan (het bedenken van oplossingen voor problemen) als het realiseren of bouwen van die oplossingen. Bij nadere beschouwing blijken tussen ontwerpen en bouwen, in de Informatica, geen wezenlijke verschillen te bestaan. Dat komt door de immateriële aard van de dingen waar informatici zich mee bezig houden. Die aard heeft tot gevolg dat realiseren of bouwen eigenlijk ook weer ontwerpen is, maar dan 'op een lager niveau'. Het doet er daarbij niet toe om welk soort systeem het gaat (bedrijfsproces, informatiesysteem of ICT-infrastructureur-component). Bij het ontwikkelen van informatiesystemen bijvoorbeeld, volgt traditioneel na functioneel en technisch ontwerp, het programmeren van de onderscheiden modules, en het opstellen van het database schema. Bij nader inzien echter zijn beide soorten activiteiten het kiezen en samenstellen van geschikte ICT-infrastructureur-componenten (denk aan objecten of routines en aan DBMS-en), en het bepalen van de juiste parameterwaarden van elke component. Van een ontwikkelaar wordt eenzelfde mate van creativiteit verwacht als van een architect, maar binnen een door de architectuur vastgelegd gebied, en daarom minder breed maar meer diepgaand. Van een ontwikkelaar wordt ook meer precisie verwacht: elk ontwerp dient te worden geverifieerd en gevalideerd. Het werk vereist dat men logisch en analytisch kan denken, en dat men daarnaast over voldoende communicatieve vaardigheden beschikt. Dat geldt met name voor de functioneel ontwerpers van informatiesystemen; zij moeten zich kunnen verplaatsen in de noden en wensen van de toekomstige gebruikers van de ontworpen systemen.

#### *Implementeren*

Implementeren is het in gebruik stellen van een ontwikkeld (en getest) systeem. Hoewel de tijd die daarmee in het algemeen is gemoeid, vergeleken met ontwikkelen of beheren, relatief gering is, is het toch als een aparte, en aan de andere evenwaardige, soort werkzaamheid opgenomen in het model. Dat komt vooral door de geheel eigen aard van het werk. Wie ooit een systeem heeft ingevoerd, weet dat dat altijd gepaard gaat met onvoorziene problemen. Dankzij een goede voorbereiding kunnen die problemen meestal snel worden opgelost, maar soms

niet. Implementeren betekent iets nieuws, en dus iets 'vreemds', in een bestaande omgeving doen functioneren, en dat is in principe een complexe aangelegenheid. Het is in dat opzicht goed te vergelijken met het vervangen van een orgaan in het menselijk lichaam door een donor-orgaan of een kunstorgaan. Dat behoort tot de meest complexe chirurgische ingrepen. Implementatoren dienen een gedegen kennis te bezitten van zowel het in te voeren systeem als van de omgeving waarin dat systeem moet gaan functioneren. Het werk vereist een hoge bestendigheid tegen stress ten gevolge van onverwachte problemen, die altijd binnen een vaststaand en vaak ook kort tijdsbestek moeten worden opgelost. De oplossingen hoeven niet perfect te zijn, als ze maar 'werken', en als vooral de invoering maar op tijd klaar is. Een ander argument voor een gelijkwaardige plaats van implementeren, is dat het implementeren van grote standaard-applicaties (zoals ERP-pakketten) een omvangrijke klus is, die vaak de gehele rij C (met uitzondering van het vakje C1) omvat.

### *Beheren*

Onder beheren wordt het in bedrijf houden van een ingevoerd systeem verstaan. Ook hier geldt, dat het systeem in kwestie een bedrijfsproces, een informatiesysteem of een stuk ICT-infrastructuur kan zijn. Grofweg zijn er twee vormen van beheer te onderscheiden. De ene vorm is het op elk moment werkend en beschikbaar houden van de actuele functie en constructie van het beheerde systeem. De andere vorm van beheer is, ervoor te zorgen dat die actuele functie en constructie de 'optimale' zijn. Wat de functie betreft, betekent dat, dat men veranderende wensen en eisen signaleert en beoordeelt, en dat men zorgt voor het aanbrengen van die veranderingen. Kleine veranderingen brengt de beheerder meestal zelf aan, grote veranderingen leiden tot (her)ontwikkeling. Wat de constructie van het beheerde systeem betreft, houdt beheren in, dat een systeem bij veranderingen in de operationele omgeving in bedrijf blijft, maar ook dat men zulke veranderingen initieert indien de prestaties van het geheel daardoor verbeteren.

Van een beheerder wordt verwacht dat hij of zij een gedegen kennis heeft van het beheerde systeem, van de gebruiksomgeving en van de operationele omgeving waarin het systeem werkt. Voorts dient men alert te zijn op veranderingen die zich in een van de twee omgevingen (kunnen) voordoen. Het werken als beheerder vereist daarom een voortdurende zorgzaamheid en een gedisciplineerde werkhouding.

## **5 Informatica: een dynamisch terrein**

Wie de personeelsadvertenties op het gebied van de Informatica bestudeert die sinds het ontstaan ervan zijn verschenen, ziet functies komen en gaan, en vaak nog in een hoog tempo ook. Hoe gaat dat nou in zijn werk, dat ontstaan en verdwijnen van functies/beroepen? Naar mijn mening spelen daarbij drie mechanismen een rol: specialisatie, concentratie en innovatie. Laat ik om het mechanisme van specialisatie te illustreren het beroep van programmeur als voorbeeld nemen. In de loop van de tijd zijn onder andere de volgende specialisaties daarvan ontstaan: Algol-, Fortran-, Cobol-, PL/I-, Pascal-, Modula-, Prolog-, C- en Java-programmeur. Het *specialisatie*-mechanisme is belangrijk in de Informatica, maar het komt ook in andere beroepsgebieden voor (denk maar aan de banketbakker en de ecologische slager). Het begrip specialisatie houdt in, dat er ook nog iets gemeenschappelijks is. Het

gemeenschappelijke van alle programmeur-specialisaties is gelukkig aanzienlijk. Voor wie het vak van programmeur ooit (goed) heeft geleerd, is het niet zo'n grote kunst in een nieuwe taal te leren programmeren (ik vergeet gemakshalve het natuurlijk niet te verwaarlozen onderscheid in procedurele, declaratieve en functionele talen).

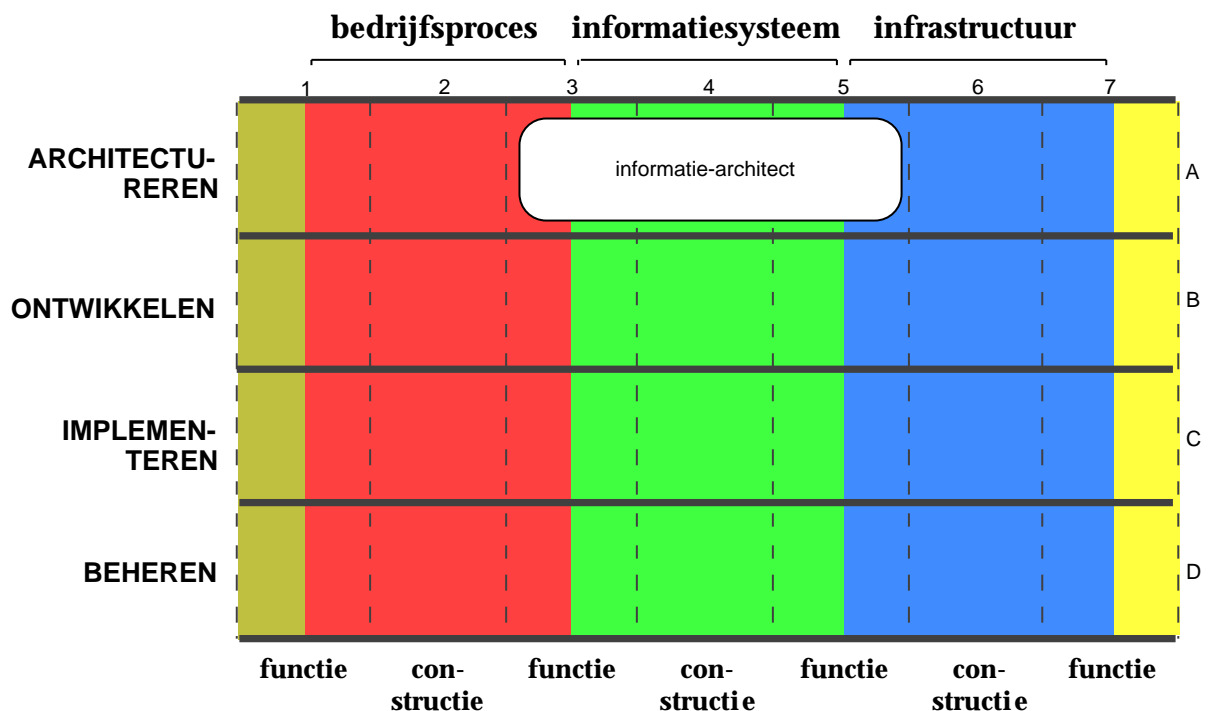
Met het mechanisme van *concentratie* wordt bedoeld dat een beroep in de loop der tijd zoekt naar zijn kern en zich daarop concentreert. Als voorbeeld neem ik weer het beroep van programmeur. Dertig jaar geleden was een programmeur iemand die het werk deed van wat nu een informatie-analist, een systeemontwerper en een codeur bij elkaar doen. In figuur 1 betekent dat, dat het domein van de programmeur de vakken B3, B4 en B5 omvatte. Nu is het iemand die uitsluitend een gedetailleerd technisch ontwerp omzet in programmacode, en dus geheel binnen het vak B5 valt. De aanname in L\_PASO is, dat de granulariteit van de structuur voldoende fijn is, dat wil zeggen dat elk (geconcentreerd) beroep een eigen domein (van een of meer vakken) heeft. De toekomst moet uitwijzen of dat inderdaad het geval is.

Het mechanisme van *innovatie* tenslotte, zorgt ervoor dat er een 'echt' nieuw beroep kan ontstaan als gevolg van het anders dan voorheen aanwenden van technologie, of van het op een geheel andere wijze concipiëren van wat men doet. Twee illustratieve voorbeelden van de werking van het innovatie-mechanisme zijn de twee grote keerpunten in het denken over informatie- en communicatietechnologie. Het eerste deed zich rond 1970 voor. Tot die tijd werd ICT vooral of uitsluitend ingezet voor de verbetering van bestaande werkwijzen, de aandacht was hoofdzakelijk gericht op de efficiënte codering van 'vormen', zoals woord- en getalrepresentaties in bits en bytes, en op de efficiënte verwerking daarvan. Anders gezegd, ICT werd ingezet als *substituut* voor de tot dan toe gebruikte technologie. De verandering in het denken is geweest, dat daarna de hoofdaandacht kwam te liggen op de betekenis die door de vormen werd vastgelegd. Door de bewuste scheiding van inhoud en vorm, ofwel van het wat en het hoe, werd de weg gebaad voor de *innovatieve* aanwending van ICT, en onstonden er nieuwe functies en beroepen, bijvoorbeeld die van informatie-analist en database administrator. Het tweede keerpunt is van zeer recente datum, en misschien is het beter te stellen dat die zich nog aan het voltrekken is. De aandacht is nu gericht op het waarom van het wat: waartoe dient informatie en communicatie, waar gaat het eigenlijk om? Men zou ook kunnen stellen dat de wat-hoe-vraag opnieuw aan de orde is, maar dan op een 'hoger' niveau. De innovatieve aanwending van ICT betreft nu de bedrijfsprocessen zelf. Voorbeelden van nieuwe functies en beroepen zijn Business Process Modeller en bedrijfsproces-analist. Afgebeeld op figuur 1, zijn er zo in de loop der tijd drie grote domeinen ontstaan: de kolommen 1,2 en 3 tezamen, de kolommen 3,4 en 5 tezamen en de kolommen 5,6 en 7 tezamen. In moderne terminologie zouden die respectievelijk Business Systems Engineering, Information Systems Engineering en Infrastructural Systems Engineering genoemd kunnen worden.

## 6 De informatie-architect

En nu dan de hamvraag: wat is het domein van de informatie-architect? Tegen de achtergrond van de hiervoor geschetste evolutie van het werkterrein Informatica, kan men zeggen dat de kiem van het beroep al na het eerste keerpunt is gevormd,

maar dat de praktische behoefte op grote schaal zich pas nu, op het tweede keerpunt, aandient. In elk geval zie ik het beroep van informatie-architect als het product van de werking van het innovatie-mechanisme. Dat maakt dat het domein van de informatie-architect nog niet duidelijk vastligt, en ook dat het nog aan verandering onderhevig zal zijn. Concentratie en specialisatie moeten hun werk nog doen, en dat zal zeker gaan gebeuren, ook al weet ik nog niet hoe. In figuur 2 is aangegeven wat naar mijn mening het (huidige) domein van de informatie-architect is.



Figuur 2 Positionering van de informatie-architect in L\_PASO

Wat betekent die positionering voor de inhoud van het werk van de informatie-architect? Om te beginnen ligt het domein in zijn geheel in de rij Architectureren. Karakteristiek voor het werk is dus dat het gaat om grote lijnen, kaders, concepten, plannen etc. De informatie-architect is dus bijvoorbeeld goed te vergelijken met de stedenbouwkundig ontwikkelaar. Het gaat om het bepalen van de kaders van de informatievoorziening binnen een organisatie. Het ontwerp van afzonderlijke informatiesystemen behoort er niet toe, dat ligt op de rij Ontwikkelen. Vervolgens strekt het domein zich uit over drie kolommen, twee 'boundary spanning'-kolommen (3 en 5) en één 'home keeping'-kolom (4). De laatste beschouw ik als de kern van het domein: wat een informatie-architect maakt, is een informatiesysteem-architectuur. Om dat goed te kunnen doen moet die architectuur 'ter linkerzijde' passen bij de bedrijfsproces-architectuur en 'ter rechterzijde' bij de infrastructuur-architectuur. Daarvoor is het nodig dat de informatie-architect grondig op de hoogte is van de denkkaders aan beide zijden. Actueel betekent dat bijvoorbeeld dat hij/zij vertrouwd moet zijn met het objectgeoriënteerde denken en de client-server-architectuur aan de kant van de infrastructuur, en met principes van logistieke besturing, en het nieuwe communicatieve-actie-paradigma aan de kant van de

bedrijfsprocessen. Tot zover mijn positionering, nu de discussie, en hopelijk gauw ook de opleidingen!

## Referenties

- 1 *Geen toekomst zonder informatica*, rapportage van de verkenningscommissie Informatica, juni 1996, ISBN 9069841517.
- 2 *De structurele behoefte aan informatici*, advies van de Adviesraad voor het Wetenschaps- en Technologiebeleid, nr. 31, februari 1998, ISBN 9034635279.
- 3 *Nationale Onderzoeksagenda Informatica*, een uitgave van SION (Stichting Informatica Onderzoek Nederland), december 1997.
- 4 *VRI-model, versie 3*, een uitgave van de Nederlandse Vereniging van Registerinformatici, oktober 1997.