

Drs. C. M. Malherbe en Dr. A. H. M. Schrama

Het gebruik van de computer bij de ontwikkeling van informatiesystemen

Synopsis

In dit artikel worden enkele problemen besproken, waarmee de praktijk zich geconfronteerd ziet, indien men de computer wil gaan gebruiken voor het ontwikkelen van informatiesystemen.

Tevens wordt aandacht besteed aan de centrale vraag, welke eis de ontwerpmethodologie stelt aan het gebruik van geautomatiseerde hulpmiddelen tijdens het gehele traject van de systeemontwikkeling.

Over de inzet van de computer als gereedschap voor systeemontwikkeling wordt reeds jaren nagedacht. Afgezien van enkele toepassingen op deelterreinen vindt 'computer aided system development' nog steeds niet op grote schaal plaats, ondanks de potentiële verhoging van effectiviteit en efficiency van de totale systeemontwikkeling. De knelpuntfactoren, die een doorbraak verhinderen, zijn echter van algemene, niet van specifieke aard; reden waarom in dit artikel wordt ingegaan op de algemene problematiek.

1. Inleiding

Voor het analyseren, ontwerpen en concretiseren van al dan niet geautomatiseerde informatiesystemen is een grote verscheidenheid aan methoden, technieken en hulpmiddelen beschikbaar. Ook de computer zelf kan onder de hulpmiddelen (of het 'gereedschap') worden gerangschikt.

Als we hier spreken van de computer als hulpmiddel denken we bijvoorbeeld aan geautomatiseerde documentatie, interactieve programma-ontwikkeling, programmatuur voor schermopmaak, data dictionaries, report generators, computer graphics, proto-typing, programma-generatoren, database design aids (zoals geautomatiseerde analyse van de gebruikerswensen en de geautomatiseerde definitie van de data base structuur), ontwerp evaluatie methoden (zoals statistische analyses), simulatie met behulp van de computer, tekstverwerking, retrieval functies en faciliteiten voor structurering, ordening en samenvatting van opgeslagen gegevens.

Indien er waarheid schuilt in de vermeende voordelen van het gebruik van de computer als hulpmiddel op het vlak van de effectiviteit en de efficiency, is het zaak deze voordelen op hun mérites te onderzoeken. Daarnaast dienen ook de werkelijke probleemfactoren die de opmars van de computer als gereedschap voor de systeemontwikkeling vooralsnog hebben belem-

merd, opgespoord te worden. Dit artikel is bedoeld als bijdrage daartoe, waarbij we ons op enkele problemen zullen concentreren: de methodologische factor, andere omgevingsfactoren en de tegengestelde opvattingen met betrekking tot de toepassing van de computer.

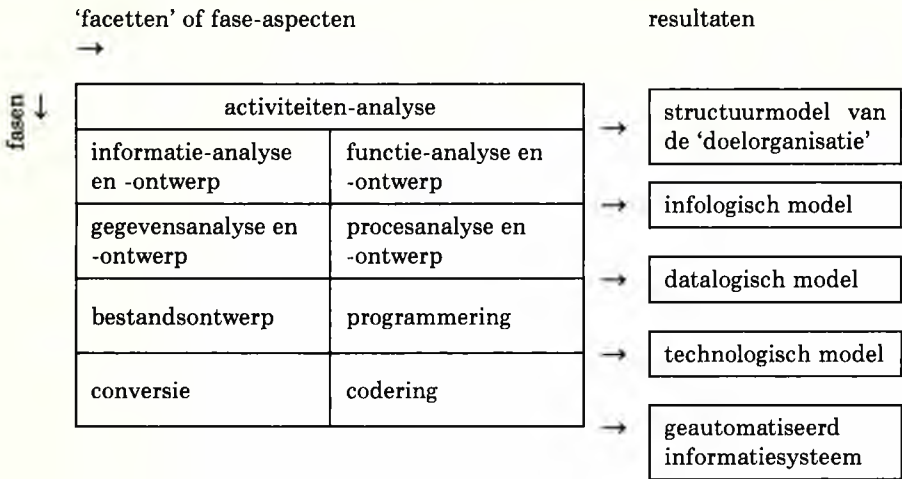
2. De methodologische factor

Wij kunnen ons eerst afvragen welke functies de computer als hulpmiddel in het totale traject van de systeemontwikkeling kan vervullen. Deze functies worden gedictieerd door een constellatie van factoren waarvan de *methodologie* evenwel zonder meer de belangrijkste is. Concentreren we ons daarom op de 'methodologische factor' dan zien we dat deze factor *beperkingen* stelt aan de feitelijke technische uitwerking van een concept van computerondersteuning. De methodologie bepaalt immers de *vorm* en de *inhoud* van systeemontwikkelingsactiviteiten, waarbij een adequaat ondersteuningsconcept moet worden aangepast dat in dit geheel bepaalde functies kan vervullen. De vraag is nu *in hoeverre* de methodologie de mogelijkheden beperkt van computer-ondersteuning. Is die onderlinge afhankelijkheid zo groot, dat er slechts één tool concept (zie par. 4) 'past' bij één methodologie, of is er meer speelruimte? Indien er speelruimte is, hoe ver gaat deze dan? Met andere woorden welke *ficties* zijn er? Welke flexibiliteit in de aanpassing is daarentegen mogelijk, en welke vrijheden worden daarbij gelaten, onder de voorwaarde dat geen sub-optimale effectiviteit en efficiency worden bereikt?

2.1. Het methodologische model

'Methodologie' betekent: 'Hoe gaat men te werk en waarom'. En daarover kan men twisten. Niet iedereen heeft dezelfde beweegredenen om te doen wat hij doet. Toch hebben de inzichten van de laatste jaren op het gebied van de informatiesysteemontwikkeling geleid tot een vrij algemeen aanvaard 'model' van ontwikkeling. Ook het te concipiëren informatiesysteemmodel heeft strakkere vormen aangenomen. Het onderstaande - zeer gestileerde - schema is ontleend aan de opvattingen van de ISAC-methode van Lundeberg (1978), doch zou evengoed ook voor andere 'methodologieën' kunnen gelden: van boven naar beneden staan de *fasen* in de ontwikkeling, elk resulterend in een nieuw 'niveau-model' van het te ontwerpen systeem, terwijl telkens parallel 'gewerkt' wordt aan onderscheiden *facetten* van het systeemmodel.

Dit - globale - ontwikkelingsmodel zal weinig beperkingen stellen aan de computer als hulpmiddel. Te gebruiken documentatiehulpmiddelen, interactieve hulpmiddelen en eventuele generatoren kunnen - indien het technisch nut daarvan bewezen is - methodologisch zonder frictie in dit model worden ingepast.



Anders wordt het bij een gedetailleerde invulling van de diverse analyse- en ontwerpactiviteiten. In ISAC maar ook in de beschrijving van andere methoden (zie bijvoorbeeld SDM van Hice e.a., 1978 en ARDI van Hartman e.a., 1972) worden deze onderverdelingen opgesomd.

De mogelijkheid om bij deze gedetailleerde activiteiten geautomatiseerde hulpmiddelen in te zetten hangt dan af van de specifieke 'geschiktheid' van zo'n hulpmiddel voor die bepaalde activiteiten. Een dergelijke afhankelijkheid bestaat ook indien men een geautomatiseerd hulpmiddel wenst te combineren met documentatie- en/of vastleggingstechnieken, waarbij een bepaalde specifieke 'wijze van afbeelden' wordt voorgeschreven (zoals bijvoorbeeld SADT en HIPO, beschreven door o.a. Blokdijk, 1981; vergelijk echter ook ISAC): de geautomatiseerde documentatie moet dan overeenkomen met de standaard voorgeschreven documentatiewijze van de gekozen methodologie.

Een bijzonder aspect vormt de eventuele automatisering van analyse- en ontwerpbehandeling *zelf*, als onderdeel van de generatorgedachte (zie par. 4.3). Veelal wordt in dat verband gesteld dat de systeemontwikkeling in zekere mate 'gestroomlijnd' zou moeten zijn wil er van 'automatisering' sprake kunnen zijn. Die opvatting geldt goed beschouwd echter ook voor het nog in par. 4.3 te bespreken 'support' concept: op de momenten en plaatsen, waar machine-mens interactie plaats vindt moet dit volgens vastomlijnde principes en afspraken gebeuren.

2.2. Stroomlijn en formalisering

Een gestroomlijnde systeemontwikkeling lijkt in tegenspraak met de gangbare projectmatige aanpak bij het ontwikkelen van informatiesystemen. Wel is samenwerking tussen mensen vereist die volgens vaste afspraken en procedures geacht worden te werken, doch dit impliceert niet dat de systeemontwikkeling één gestroomlijnd geheel van activiteiten is, zelfs niet wat fase-onderdelen aangaat. Ook het te concipiëren model van het infor-

matiesysteem, zijnde het doel van de ontwikkeling, is niet gestroomlijnd. Integendeel, het is juist het object van verandering, aanpassing en modificaties.

Hoe kan de stroomlijn-eis, die toch ten grondslag lijkt te liggen aan het gebruik van geautomatiseerde hulpmiddelen, dan niettemin worden gerealiseerd in het kader van de systeemontwikkeling? Om die vraag te kunnen beantwoorden moeten we het accent verleggen van 'stroomlijn' (overigens een niet operationeel te maken begrip) naar *vormgeving*. *Vormkenmerken* zijn operationeel in de communicatie tussen mensen onderling, echter ook in de mens-machine interactie. Het is daarom de mogelijkheid tot *formalisering* van ontwikkelingsactiviteiten die bepalend is voor de mate waarin met succes geautomatiseerde 'tools' kunnen worden gebruikt. Formalisering betekent in onze visie: er wordt voldaan aan bepaalde 'afgesproken' vormkenmerken.

De communicatie van de vormkenmerken staat centraal, niet de stroomlijn van de ontwikkelingsactiviteit. Nauwkeurig omschreven ontwerpstappen binnen een methodologie zijn dan ook geen beletsel voor computer-ondersteuning, noch vormen zij daartoe een voorwaarde. Wat telt zijn de afgesproken vormkenmerken die inherent zijn aan de omschreven activiteiten.

2.3. De geautomatiseerde documentatie

Voor het 'computer aided development' heeft het formaliseringsaspect van de methodologie de consequentie, dat de nadruk komt te liggen op de *documentatie*, alsook op het manipuleren met gegevens, die reeds in de documentatie zijn vastgelegd. Methodologieën die dan ook het beste met automatische hulpmiddelen gecombineerd kunnen worden, zijn ons inziens deze welke de nadruk leggen op de hoofdfunctie die de documentatie in de systeemontwikkeling vervult, namelijk op de *communicatie*: de communicatie tussen systeemontwerpers/programmeurs onderling, tussen systeemontwerpers en 'eindgebruikers', en tussen eindgebruikers onderling. De geautomatiseerde documentatie berust dan op twee pijlers:

- a) de machinale vastlegging van (tussen-)resultaten van de systeemontwikkeling,
- b) de presentatie van deze gegevens ten behoeve van de diverse communicatie-processen.

De ISAC-methode bijvoorbeeld legt in hoge mate de nadruk op de communicatie; dit blijkt ondermeer uit het gebruik van de visuele activiteiten-schema's e.d.. Indien hierbij een geautomatiseerde documentatietechniek zou kunnen worden gebruikt, zou het communicatie-aspect van ISAC zeer worden versterkt, waardoor de effectiviteit en de doelmatigheid van deze methode zouden toenemen. Bij een andere methode, namelijk SMX (zie Schell, 1981) is de hier geschetste uitwerking van geautomatiseerde ondersteuning reeds in de praktijk gebracht.

Een uitzonderlijk voordeel van 'communicatie-tools' is (naast het sub a genoemde) dat de presentatie - sub b - snel en efficiënt kan geschieden *in de voor de gebruiker meest bruikbare vorm*. Bijvoorbeeld voor een programmeur: beslissingstabellen en I/O-tabellen; voor een 'eindgebruiker': informatie-stroomschema's en 'formulieren'. Voor de machine is de meest effi-

ciënte vorm uiteraard de machinale representatie waarin de vastlegging geschiedt.

3. Andere omgevingsfactoren

We zagen dat het traject van de systeemontwikkeling uiteenvalt in diverse vaak totaal verschillende fasen. Binnen elk van deze fasen zijn meestal meerdere soorten activiteiten te onderkennen: functiegerichte, procesgerichte, controlegerichte en gegevensgerichte activiteiten.

De vraag naar de praktische mogelijkheden van de inzetbaarheid van de computer binnen elk van deze fasen en activiteiten vereist uitspraken omtrent de gewenste of noodzakelijke graad van horizontale dan wel verticale integratie, welke binnen het raam van het *gehele* traject kan of moet worden gehanteerd. Met horizontale integratie bedoelen we hier de integratie *binnen een fase* van het traject der systeemontwikkeling, met verticale integratie die *tussen* de verschillende fasen. Zonder een eenduidige uitspraak dienaangaande blijft de vraag naar de mogelijkheid van computergebruik binnen het *gehele* traject (en die vraag willen we hier centraal stellen) naar onze mening onbeantwoord, hetgeen een zwakke en dus onwerkbare uitgangsstelling tot gevolg zou hebben, die gedoemd is ten eeuwigden dage als puur theoretische mogelijkheid te blijven voortsluimeren. Binnen bepaalde fasen(onderdelen) wordt de computer overigens reeds geruime tijd als hulpmiddel toegepast. Denk hierbij aan data dictionaries, computer graphics en programma-generatoren.

De integratiemogelijkheden zijn op hun beurt bepalend voor de praktische mogelijkheden van computerondersteuning per afzonderlijk geval van systeemontwikkeling. De complicerende factor daarbij is niet zozeer dat elk concreet project van nature verschillend van aard, omvang, reikwijdte en diepgang is, doch eerder dat elk project beheerst wordt door een aantal specifieke beperkingen van *organisatorische*, *methodologische* (vorige par.), *terminologische* en *technologische* aard, de zogenaamde omgevingsfactoren. De aansluiting ('fit') van een bepaald concept van computerondersteuning, die bij een bepaalde constellatie van de genoemde factoren verkregen kan worden, zal daardoor in vele gevallen niet hoeven op te gaan voor een andere samenstelling van factoren. Voor de praktijk betekent dit dat in elk specifiek geval een keuze gemaakt moet worden uit een van de volgende opties:

1. zoeken naar een reeds bestaand ondersteuningsconcept dat past bij de aanwezige constellatie van factoren,
2. zelf een passend ondersteuningsconcept ontwikkelen,
3. indien mogelijk, streven naar een wijziging van de factoren zodat een 'fit' verkregen wordt met het gewenste ondersteuningsconcept,
4. afzien van het gebruik van de computer bij de systeemontwikkeling.

De genoemde mogelijkheden zijn typerend voor de huidige praktijk. De bestaande kloof tussen de vraag naar en het aanbod van een in concreto passend systeem van geautomatiseerde ondersteuning van de ontwikkeling van informatiesystemen lijkt de noodzaak aan te geven van een *universeel* (d.w.z algemeen toepasbaar) geautomatiseerd hulpmiddel. Het dilemma dat

zich daarbij echter voordoet is dat een dergelijk universeel concept nog niet het meest doeltreffende en doelmatige behoeft te zijn, met andere woorden er kunnen andere - niet universele - concepten zijn die in specifieke gevallen een betere 'fit' bewerkstelligen. En dit dan nog afgezien van de vraag of een universeel systeem enige kans maakt in de nabije toekomst tot ontwikkeling te worden gebracht. Momenteel bestaat zo'n algemeen systeem voor het gehele ontwikkelingstraject nog niet, al zijn er enkele vermeldenswaardige pogingen in die richting gedaan. Het is een open vraag of het ooit zo ver zal komen. De factoren die bij de systeemontwikkeling een rol spelen zijn dermate complex en dynamisch dat 'universaliteit' een wel zeer moeilijk in te bouwen eigenschap is. Niettemin zullen er ten aanzien van 'deelproblemen' in de systeemontwikkeling wel degelijk grote resultaten geboekt gaan worden. Enkele ontwikkelingen te dien aanzien zijn veelbelovend en zullen ongetwijfeld de komende jaren meer en meer toepassing vinden. Ook om die ontwikkelingen te volgen heeft het zin nader in te gaan op enkele achtergronden van het 'computer aided development'.

4. Enkele 'hete hangijzers'

In de gedachtenvorming over de computer als ondersteuningsmiddel ten behoeve van de ontwikkeling van informatiesystemen zijn er diverse voorstanders die zich op een bepaalde benadering willen vastleggen. Anderen die een andere benadering voorstaan wensen echter van hun standpunt vooralsnog niet te wijken. De geschilpunten lijken voorlopig onoverbrugbaar. Hieronder wordt op enkele van die geschilpunten nader ingegaan.

4.1. De visie ten aanzien van de levenscyclus

Teichroew (1972) spreekt van de 'system life cycle' als continu proces van systeemontwikkeling. In zijn opvatting bestaat deze uit de volgende fasen:

- system planning
- logical system design
- physical system design
- system construction
- system test/conversion/installation
- operation
- maintenance and modification.

Zijn concept van 'computer aided structured documentation and analysis of information processing system requirements', beter bekend onder de naam ISDOS (Information Systems Design and Optimization System) sluit daar dan ook geheel bij aan. De implicatie is dat wie ISDOS gaat gebruiken ook wordt uitgehuwelijkt aan de daaraan ten grondslag liggende opvatting ten aanzien van de levenscyclus. Een andere fasering zal - zo men deze wenst - onder het regime van ISDOS moeilijk zijn te verwezenlijken. Wie kiest voor ISDOS kiest voor Teichroew's denkwijze en voor de daaraan verbonden werkwijze. Een kanttekening is hierbij op zijn plaats:

Denkwijze en werkwijze behoren een 'methodologische eenheid' te vormen, doch dit betekent niet dat een flexibel systeem van computerondersteuning (lees: een bepaalde werkwijze) niet dienstbaar zou kunnen worden gemaakt

aan verschillende opvattingen ten aanzien van de levenscyclus en aan daarmee verbonden denkwijzen.

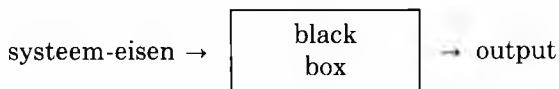
4.2. 'Tool' versus 'instrument' concept

Sommigen beschouwen de computer uitsluitend als *hulpmiddel* voor systeemanalyse, -ontwerp, programmering, e.d. Vaak kunnen er meerdere 'tools' zijn, bijvoorbeeld in de vorm van verschillende programma's die, al dan niet op één ontwikkelingscomputer, tezamen een aaneensluitend geheel vormen. Een dergelijke groep van 'tools' die binnen een raamwerk van een bepaalde methodologie hun aanvullende en ondersteunende diensten verrichten rekent men tot het zogenaamde 'tool concept'. Inherent aan het 'tool concept' is de gedachte dat geautomatiseerde hulpmiddelen pas tot dat concept worden gerekend als zij voldoen aan methodologische criteria behorende bij dat concept. Hulpmiddelen die niet aan die criteria voldoen worden niet tot het concept gerekend. Wij moeten hier echter oppassen, want één van de methodologische criteria die men aan een 'tool concept' zou kunnen verbinden, zou juist kunnen zijn dat het concept in *elke* methodologie toepasbaar moet zijn. Op grond van zo'n criterium zouden alle 'tools' in elk methodologisch kader gebruikt kunnen worden.

Wil men daarentegen de computer als het *enige* middel gebruiken om een bepaald fase-onderdeel van de systeemontwikkeling te realiseren dan heeft het ook geen zin te zoeken naar aanvullende 'tools' die onder een vigerend tool-concept zouden kunnen worden gerekend. Bij het 'tool concept' kan nog worden gekozen uit meerdere 'tools' (passend binnen het gekozen methodologische concept). Deze keuze is nu niet meer mogelijk en we spreken hier van het 'instrument concept'. Elke aanvulling is overbodig en draagt uit dien hoofde niet bij tot de effectiviteit en de efficiency van de ontwikkelingsactiviteiten. In dit geval is er dan ook geen keuzeproblematiek zoals bedoeld in par. 3. In de volgende paragrafen zullen we voornamelijk het meer voor de hand liggende 'tool concept' in beschouwing nemen.

4.3. 'Generator' versus 'support' concept

Een onderscheid dat enige gelijkenis vertoont met het hiervoor behandelde, maar dat toch van fundamenteel andere aard is, is dat tussen de 'generatorgedachte' en de 'ondersteuningsgedachte'. De generator stoelt in essentie op een 'black box' benadering:



Daarbij is de gedachtengang dat alle activiteiten die in een bepaald fase-onderdeel anders handmatig met behulp van adequate 'tools' moeten worden verricht nu geheel door de 'black box' kunnen worden uitgevoerd. De bedoelde activiteiten kunnen zowel analytisch als ontwerptechnisch van aard zijn, dan wel betrekking hebben op programmering of conversie. De black box is *generatief*: na invoering van alle vereiste gegevens wordt geheel

automatisch een resultaat gegenereerd. Voorbeelden van generatoren zijn: pre- of macroprocessorsen voor de programmering, report writers, generatoren voor beslissingstabellen, etc.

De ondersteuningsgedachte daarentegen sluit aan bij het 'tool concept': zodra een 'tool' geschikt is om binnen het raam van de methodologie ondersteunende diensten te verrichten kan deze in principe worden aangewend. 'Supporting' vereist - in tegenstelling tot het generatorconcept - *interactie* tussen machine en mens (analist, ontwerper, programmeur).

4.4. 'Opportunity' versus 'permanent' concept

De selectie (eventueel: ontwikkeling) van geautomatiseerde hulpmiddelen ten behoeve van een 'tool concept' brengt onder meer met zich mee dat bekeken moet worden over welke periode en voor welke projecten het gekozen hulpmiddel is in te zetten. Men spreekt van een 'opportunity-concept' als een bepaald hulpmiddel voor slechts eenmalig gebruik is te gebruiken. Betreft men in zijn overwegingen het feit dat de systeemontwikkelingscyclus in wezen een continu, zich zelf herhalend karakter heeft, dan kan dit ertoe leiden dat men zal besluiten een 'permanent' tool concept te realiseren, waarop men telkenmale kan terugvallen. Zulks geldt met name voor bijvoorbeeld software-bureaus en ontwikkelingskantoren die van de systeemontwikkeling hun dagelijkse praktijk hebben gemaakt. Ook voor 'in-house' projecten echter kan men tot een dergelijk besluit komen en wel in het bijzonder ten aanzien van documentatie- en programmeringshulpmiddelen.

4.5. Doelcomputer versus hulpcomputer

Een volgend probleem is dat van de keuze tussen een tool-systeem dat 'draait' op de computer van het nog te ontwikkelen informatiesysteem enerzijds (doelcomputer) en een apart systeem met een eigen ontwikkelingscomputer anderzijds. Men spreekt hier ook wel van 'host' versus 'stand-alone' uitvoering. Een factor die mede bepalend is voor deze keuze zijn de kosten, waaronder de kosten van de noodzakelijke conversie van ontwikkelingscomputer naar doelcomputer. Deze kosten kunnen betrekkelijk laag dan wel zeer hoog zijn, afhankelijk van de overdraagbaarheid (portabiliteit) van de door het ontwikkelingssysteem opgeleverde programmatuur en bestandsstructuren. Hoe hoog de kosten zullen zijn is echter vooraf niet precies te zeggen omdat niet bekend is hoe programma's en gegevensmodellen er uit zullen zien. Heeft de ontwikkeling plaats op de doelcomputer dan is het voordeel in ieder geval dat er niet meer geconverteerd hoeft te worden. Een nadeel kan zijn gelegen in additioneel noodzakelijke wijzigingen in capaciteit, configuratie e.d. van het doelsysteem (als gevolg van de resultaten van de systeemontwikkeling).

Een belangrijke conclusie is dus dat de keuze tussen ontwikkeling op aparte hulpcomputer of direct op de doelcomputer in het algemeen wordt vergemakkelijkt naarmate:

1) de portabiliteit van de op te leveren bestandsstructuur en programma-tuur hoger is;

2) er meer faciliteiten voor conversie zijn.

Hoe meer aan beide factoren wordt voldaan des te meer komt bovendien *elk* stand-alone systeem in aanmerking, dit ten nadele van de 'host'-gedachte. Als doorslaggevend extra voordeel van een stand-alone benadering wordt algemeen erkend dat een dergelijk ontwikkelingssysteem qua configuratie en technische uitwerking geheel kan worden toegesneden op de functies die men aan het ontwikkelingssysteem wil toekennen.

Naast documentatie- en genereringsfuncties kan bijvoorbeeld een zogenaamde 'methodenbank' gebruikt worden. In zo'n methodenbank bevinden zich allerlei hulproutines, alsook standaard structurelementen die men tijdens analyse en ontwerp kan gebruiken.

5. Slotopmerkingen

De mogelijkheden van de computer als gereedschap bij de systeemontwikkeling zijn zeker niet beperkt tot de geautomatiseerde documentatie, hoewel daarin zonder twijfel het sterkste punt ligt van welk 'tool concept' dan ook. Een flexibel 'switchen' tussen machinale vastlegging en de meest doelmatige vorm van informatie-presentatie is een van de krachtigste vormen van ondersteuning die denkbaar is. De diverse in de praktijk gebruikte methodologieën pogen een zekere 'stroomlijn' aan te brengen in het ontwikkelingstraject. Toch is dat naar onze mening geen primaire voorwaarde voor het gebruik van effectieve geautomatiseerde technieken. Andere benaderingen die binnen een bepaalde methodologisch gefundeerde aanpak toch differentiaties toelaten en meer de nadruk leggen op de *communicatie* tussen alle betrokkenen, hebben duidelijk onze voorkeur. De te geven regels dienen dan meer het karakter te hebben van 'richtlijnen' volgens welke gedacht en gewerkt moet worden maar waarbij dan toch zekere vrijheden worden gelaten met betrekking tot gedetailleerde uitwerkingen en het gebruik van specifiek ondersteunende technieken.

De mogelijkheid van een 'universeel' (algemeen toepasbaar) tool concept is hier verder buiten beschouwing gelaten. Voor een nadere bespreking over die problematiek, alsook over de werkwijze en de techniek van enkele in de praktijk gebruikte geautomatiseerde hulpmiddelen mogen wij verwijzen naar een eerder verschenen publikatie van onze hand over dit onderwerp (Malherbe e.a., 1980). Daarin worden bovendien enige bijkomende aspecten behandeld die met betrekking tot het 'computer aided development' een rol van betekenis spelen zoals de organisatie van automatiseringsprojecten, de bedrijfsorganisatorische omgeving, de technologie en de commercie.

Tenslotte nog een opmerking over de vraag of een tool-concept uit één stuk gegoten moet worden of dient te zijn samengesteld uit verschillende onderdelen en verschillende 'legeringen' van tools met betrekking tot onderscheiden fasen en facetten van de systeemontwikkeling. In het geval van 'samenmelting' zal de zwakste schakel en het zwakste metaal bepalend zijn voor de sterkte van het totale concept. Het aaneensmeden van tools dient in ieder geval te geschieden op methodologische gronden. De mate waarin

men tools op elkaar kan laten aansluiten (*verticale integratie*) is afhankelijk van de mate waarin men de geautomatiseerde documentatie van een voorafgaande fase in de daarop volgende fase zonder conversieproblemen kan gebruiken. De *horizontale* integratie wordt dan bepaald door de mogelijkheden om in de documentatie (en in de machinale vastlegging daarvan) relaties tussen fase-aspecten (binnen één fase) aan te brengen. Naar de mate waarin men er in slaagt de hier bedoelde horizontale en verticale koppelingen aan te brengen wordt de reikwijdte van een tool-concept vergroot en nemen daardoor tevens de kansen op een meer algemeen gebruik van de computer als ontwikkelingsgereedschap toe.

Literatuur

1. Blokdijk, P., 1981, Een analyse van de mogelijkheden voor het structureren van informatiesystemen, *Informatie* 5/81, pp. 297 e.v.
2. Hartman, W., Matthes, H., Proeme, A., 1972, ARDI, Information systems handbook, N.V. Philips-Electrologica, Apeldoorn.
3. Hice, G. F., Turner, W. S., Cashwell, L. F., 1978, System Development Methodology, North-Holland Publ. Cy., Amsterdam. Zie ook: Oudshoorn, H., 1981, SDM en de technieken TIA, GOS, GOP en TOT, *Informatie* 4/81, pp. 207 e.v.
4. Lundeberg, M., 1978, A systematic approach to information systems development (Part I - Introduction. Part. II - Problem and data oriented methodology), ISAC Research Group, General Report TRITA-IBADB - 4423 1978-05-03, University of Stockholm, Sweden. Zie ook: Ruys, H. D., 1981, De ISAC-methodiek, *Informatie* 5/81, pp. 286 e.v.
5. Malherbe, C. M., Schrama, A. H. M., Joëls, E. J., 1980, Geautomatiseerde hulpmiddelen bij de ontwikkeling van informatiesystemen, Research Memorandum no. 8012, Faculteit der Economische Wetenschappen van de Universiteit van Amsterdam.
6. Schell, H. J., 1981, Systematrix (SMX), *Informatie* 6/81, pp. 361 e.v.
7. Teichroew, D., Carlson, D. M., 1972, A model of the system building process, ISDOS A., 1979, PSL/PSA, een hulpmiddel bij het ontwerpen van informatiesystemen, *Informatie Working Paper No. 64*, University of Michigan, Ann Arbor, Mi., USA. Zie ook: Elsenaar, 5/79, pp. 300 e.v. Zie ook: Gersteling, H., 1980, Het gebruik van PSL/PSA in de praktijk, de ISDOS-ontwikkelingen, ASI-Leergang Infologica, TH Delft, pp. 101-127. Zie ook: Gersteling, H., Schotgerrits, A. H. J. B., 1981, Het ISDOS-project en PSL/PSA, *Informatie* 9/81, pp. 513 e.v.