



Hardnekkig misverstand

Sinds de jaren 80 van de vorige eeuw is het langzaam duidelijk geworden, dat er aan de omvang van een computersysteem grenzen zijn. Dat zijn geen principiële grenzen maar puur praktische. De huishoudelijke werkzaamheden binnen een computertoepassing groeien namelijk ruwweg kwadratisch met zijn omvang. Dat loopt op een zeker moment spaak: het systeem wordt onhaalbaar duur.

In dit verband moet 'omvang' begrepen worden als het product van twee factoren: de uitgebreidheid van het systeem (zeg het aantal terminals) en het aantal functies van het systeem. Systemen zijn maakbaar als tenminste een van die twee factoren klein is, anders niet.

Overigens is doorgezette integratie van functies op zichzelf niet erg verstandig: dat heeft te maken met de onaanpasbaarheid van zo'n systeem. Als er iets moet veranderen, hebben we meestal maar één optie: vervangen van het hele systeem. Hoewel er tot op de dag van vandaag nog steeds 'gelovigen' rondlopen, zijn we er in grote lijnen wel met de neus op gedrukt: de grandioze oplossing, het allesomvattende, perfecte systeem bestaat niet, en we kunnen ook niet telkens van voren af aan beginnen.

Of we het nu leuk vinden of niet, of we er in geloven of niet, of we de illusie hebben dat het er ooit nog eens van komen zal, we staan thans voor de opdracht, systemen van diverse komaf, van diverse eigenaars, met diverse hardware, besturingsprogrammatuur, et cetera, et cetera met elkaar te laten samenwerken, op een automatische manier (met de hand en voeten interface kan het allang). Het ziet er naar uit dat dit probleem momenteel op vele plaatsen verschillend wordt aangepakt. Dat hoeft ons niet zo te verbazen als we inzien dat het een nieuw en moeilijk probleem betreft.

Tot mijn opperste verbijstering echter proberen sommigen de oplossing te construeren door alle betrokken (deel)systemen onder te brengen in één, alles omvattende infrastructuur. De communicatie tussen de (deel)systemen wordt geacht te verlopen door middel van een universeel protocol, standaardcodes en geüniformeerde input / output procedures! Hadden we dat al niet eens eerder gehoord? Zijn ze opnieuw vergeten dat er geen efficiënte wereldtaal kan bestaan? Hebben ze alweer niet begrepen dat standaardiseren en centraliseren op zichzelf een informatiesysteem behoeft; een informatiesysteem dat niet te veel omvang kan hebben!

Een aardige complicatie doet zich voor. Een 'universele' oplossing kan in een klein bestek wel werken, maar in een groot bestek niet. Zolang dus het systeem in het ontwerp stadium verkeert, dan wel in een aanloopfase van de bouw, blijkt de onhaalbaarheid niet. De uitgebreidheid is dan immers klein, en de functionaliteit is opgedeeld in modules. Pas gedurende de integratietest en de bulktest worden de bezwa-

ren evident. In andere woorden: zolang er AAN gewerkt wordt, is alles haalbaar en zo gauw er MEE gewerkt wordt, komen de tegenvallers als een lawine aan het licht.

Zo kan het opnieuw gebeuren dat bepaalde leveranciers van software vol vertrouwen claimen dat zij de universele oplossing voor dit soort netwerken in handen hebben. En dat die "binnenkort leverbaar" zal zijn. We zijn dit type beloftes al eerder tegengekomen. Ik heb ze ooit gek-scherend de "4 P verhalen" genoemd. De eerste P staat voor Piekeren, dat levert een ontwerp en een bouwplan op. De tweede P komt van Publiceren; je moet in een vroeg stadium klantjes werven. De derde P is van Proberen, de vierde van de doofPot waarin het project na ettelijke jaren tegenslag en uitstel terechtkomt. Zo heeft de industrie bijvoorbeeld – het is zo lang geleden dat ik die misser nu wel even uit de mottenballen mag halen – ooit een computer aangekondigd (de 360/65M multiprocessor) die in de praktijk verdronk in de voor het besturingssysteem benodigde computercapaciteit. De machine was praktisch altijd bezig om uit te vinden wie er aan de beurt was! Ook het droevige verhaal van de automatische treinbesturing bij de metro van San Francisco levert een voorbeeld van de 4 P's op: de vierde P was zo effectief dat niemand van SBIT er van gehoord zal hebben. Zo werkte het toen, zo zal het weer werken, en er zullen er weer veel instinken. ■