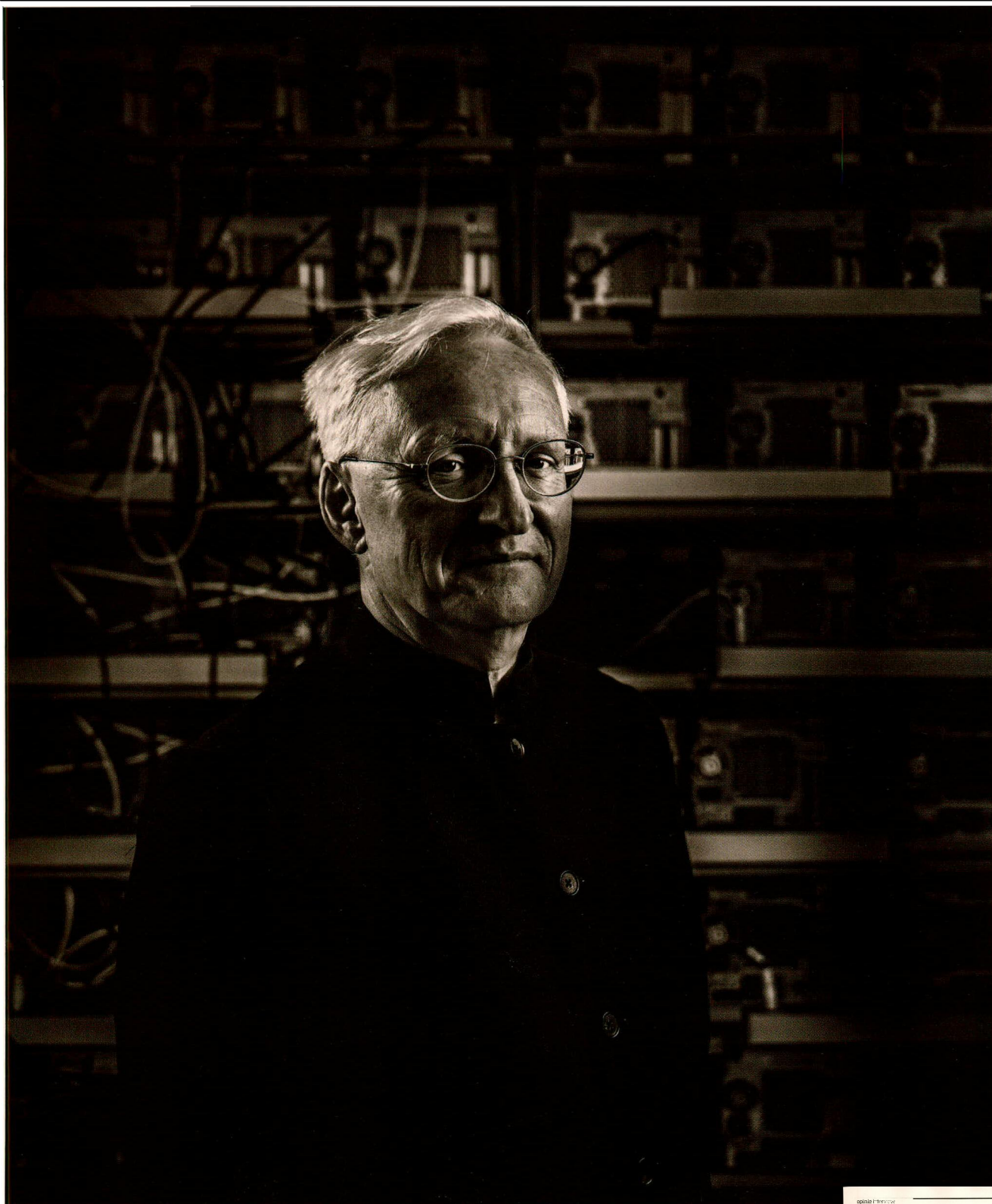


'Programmeren kan tien keer sneller'

Onze manier van programmeren stamt nog uit de tijd dat computerkracht duur was en mankracht goedkoop. Terwijl het nu precies andersom is. Tijd om over te stappen op een nieuwe programmeertaal, vindt Steven Pemberton.

Door Jop de Vrieze

De huidige manier van programmeren is zo omslachtig dat er al snel ettelijke duizenden regels code en heel veel manuren bij komen kijken. Ooit was dat logisch, want toen de taal werd uitgevonden waarin computerprogramma's worden geschreven, niet lang na de Tweede Wereldoorlog, waren computers traag en was arbeid goedkoop. Inmiddels zijn de computerchips al gauw een miljoen keer krachtiger en goedkoper, maar nog steeds wordt vrijwel alle software op deze manier geschreven. Gelukkig is er een alternatief, vertelt Steven Pemberton, onderzoeker aan het Centrum voor Wiskunde en Informatica in Amsterdam.

Waarom is de manier waarop we programmeren zo omslachtig?

'Een programmeur kauwt alles voor wat de computer onder alle verschillende omstandigheden moet doen. Voor elk detail en elke uitzonderingssituatie moet een oplossing worden ingevoerd. Terwijl dat helemaal niet meer nodig is. Door de code heel anders te formuleren, kun je veel meer van het werk aan de computer overlaten, omdat de computer vrijwel gratis is en de programmeur extreem duur. Al sinds begin jaren tachtig is de programmeur relatief duurder dan de computer.'

Hoe kan het slimmer?

'Door niet alles wat de computer moet doen tot in detail te beschrijven, maar het probleem te formuleren en de computer de oplossing te laten berekenen. Declaratief programmeren heet dat, en het format waarin we werken heet XForms. Ik leg het meestal zo uit: als

je op school zit dan leer je hoe je getallen optelt, vermenigvuldigt enzovoort. Iedereen komt van school met de vaardigheid om twee cijfers grote getallen te vermenigvuldigen. Maar niemand weet bijvoorbeeld hoe je een wortel trekt. Wat het is, dat weten we wel: een getal dat als je het met zichzelf vermenigvuldigt, je de oorspronkelijke noemer teruggeeft. Nou, dat is declaratief. Wie declaratief programmeert, beschrijft niet alle stappen die de computer moet doorlopen om een wortel te trekken, maar beschrijft wat een wortel is en laat de manier waarop die berekend wordt aan de computer over. Dat bespaart heel veel werk en bovendien heel veel bugs.'

Hoe werkt dat in de praktijk?

'Een mooi voorbeeld is de klok op je scherm. Als je als je een programma daarvoor bekijkt dat in procedurele taal is geschreven, dus op de ouderwetse manier, is het minstens duizend regels code, waarvan maar een klein stukje iets met tijd te maken heeft. Terwijl ik met declaratieve taal een voorbeeld heb van twaalf regels code. Daarbij zeg je heel simpel: wat is een klok? Het zijn drie wijzers met een cirkel eromheen en elke wijzer heeft een bepaalde lengte. Ze roteren op het aantal seconden, dus maal zes graden, want in totaal zijn dat er 360. Dat is de kern van het probleem.'

'Kauw niet alles voor wat een computer moet doen, maar laat 'm zelf de oplossing berekenen'

Zijn er al organisaties en bedrijven die ermee werken?

'In Nederland werkt het KNMI ermee en Kadaster, daarbuiten onder meer de BBC en de National Health Services, de NASA en bedrijven zoals Xerox en Yahoo.'

En wat is in de praktijk hun besparing?

'Neem een Engels verzekeringsbedrijf. De manager wilde XForms uitproberen en vroeg voor de volgende applicatie twee programmeurs een inschatting te maken hoeveel tijd ze nodig zouden hebben. Twee dagen later meldde de gewone programmeur dat hij meer tijd nodig had om de tijdsinschatting te maken. De XForms-programmeur had de applicatie al gebouwd. Een ander voorbeeld is een scheepsbouwer die software nodig had voor de controlekamers van cruiseschepen. Dat is enorm ingewikkeld en kost normaal vijf jaar met dertig mensen. Met XForms hebben ze het in een jaar met tien man gedaan. Dat zijn tien persoonsjaren in plaats van 150!'

Hoe komt het dan dat men nog niet massaal is overstapt? Zijn programmeurs bang dat ze hun baan kwijtraken?

'Klopt, niet iedereen zit erop te wachten. Niet alleen zouden er minder programmeurs nodig zijn, hun programmeerkennis is ook ineens veel minder waard geworden. Bovendien willen bedrijven die het gebruiken niet dat de concurrentie het te weten komt, omdat ze er veel kosten mee besparen. Er zijn wel softwarebouwers die vooral met XForms programmeren, maar de meeste presenteren zich niet zo. Het moet dus komen van mensen zoals ik. Ik geef daarom veel lezingen en mijn volgende stap is een boek, voor dummies, maar dan professioneel. Ik ben ervan overtuigd dat het een disruptieve technologie is: deze technologie gaat de oude vervangen en als er een bepaald punt is bereikt stapt iedereen erop over.' ■