

*Cobol 2000 uitgerust met object-oriëntatie en nieuwe
validatiefaciliteiten*

Springlevend fossiel

Cobol bestaat al meer dan veertig jaar als programmeertaal. Volgens velen is de taal ook de komende veertig jaar niet weg te denken uit de wereld van de software-ontwikkeling. Eind december van dit jaar komt het objectgeoriënteerde Cobol 2000 uit. 'Old soldiers never die'.

Cok de Zwart in gesprek met Wim Ebbinkhuijsen, lid van de illustere Amerikaanse Cobol-commissie.

Met stijgende verbazing hoorden de dagvoorzitter en de zeventig bezoekers van het congres Cobol-Vision dat in december werd gehouden, de voordracht aan van Harry Sneed, een Engelse Cobol-deskundige. We moeten maar heel snel stoppen met Cobol, luidde zijn boodschap. De programmeertaal is volstrekt achterhaald en niet meer van deze tijd. Er zijn geen jonge mensen meer bereid om de taal te leren.

Sneed is een vermaard Cobol-kenner en re-engineering consultant bij Case Consult. Hij sprak onder meer over de toevoeging van object-oriëntatie aan Cobol. Sneed is in de loop van zijn carrière een beetje gefrustreerd geraakt. Een man van tegen de 60 jaar die zijn leven lang zijn hele ziel en zaligheid heeft gegeven aan Cobol, maar ziet dat steeds minder mensen in zijn omgeving nog het enthousiasme voor de taal kunnen opbrengen. Hij beschouwt zich als een der laatste roepende Mohikanen in de woestijn. Vandaar zijn emotionele verhaal waar niemand op had gerekend.

De vraag is natuurlijk of Sneed gelijk heeft: is Cobol een ouderwetse taal die haar langste tijd heeft gehad? Nee, luidt het antwoord van al degenen die objectief en met enige kennis van zaken naar de wereld van de software-ontwikkeling kunnen kijken. Zolang er mainframes bestaan, zullen er ook Cobol-applicaties blijven draaien. De meerderheid van de bedrijfskritische systemen die vandaag de dag operationeel zijn, draait op mainframes en zijn in Cobol geschreven. Meer dan negentig procent van de Fortune 500-bedrijven heeft een mainframe in hun rekencentra staan. Geschat wordt dat al die systemen dagelijks zo'n dertig miljard transacties verwerken.

Het is duidelijk dat deze systemen gekoesterd moeten worden. Temeer daar al die mainframe-gebruikers aan het einde van de vorige eeuw miljarden guldens hebben uitgegeven om het jaar-2000 probleem op te lossen. Tegelijkertijd is de programmatuur opgeschoond en aangepast aan de huidige eisen, zodat de applicaties weer jaren mee kunnen. Bijzonder weinig organisaties hebben het

jaar 2000 aangegrepen om het mainframe de deur uit te doen en de programma's voor een ander hardwareplatform te herschrijven. Een duidelijk bewijs dat die systemen nog steeds tot de kostbaarste bezittingen van een organisatie behoren. Uit kostenoverwegingen was en is het nog steeds voordeliger om de systemen te onderhouden dan om over te stappen op iets geheel nieuws. Dat was de boodschap tenminste die Chris Verhoef, hoogleraar Computer Science aan de Vrije Universiteit in Amsterdam, op het Cobol-Vision-congres uitdroeg.

Validatiefaciliteiten

In 1959 werd op initiatief van een aantal grote gebruikers van computersystemen, waaronder de Amerikaanse overheid, Codasyl (Committee on Data Systems and Languages) in het leven geroepen. Deze organisatie werd belast met de opdracht een taal te ontwerpen die geschikt moest zijn voor het programmeren van administratieve toepassingen. Bovendien moest de taal een standaardopbouw hebben en universeel toepasbaar zijn voor alle computers. In januari 1960 werd de eerste versie van de taal gepresenteerd onder de naam Cobol, een acronym voor Common Business Oriented Language, later Cobol-60 genoemd. In de loop der jaren is de taal verder uitgebreid en aangepast, telkens vastgelegd in nieuwe standaards, zoals Cobol-68 en Cobol-74. In 1985 werd de recentste versie gepubliceerd onder de naam Cobol-85. Als alles gaat zoals het moet gaan, komt in december de nieuwste release uit: Cobol-2000. Inderdaad, twee jaar later dan gepland. De dagvoorzitter van Cobol-Vision, Wim Ebbinkhuijsen heeft, als lid van de illustere Amerikaanse Cobol-commissie een belangrijke bijdrage geleverd aan het tot stand komen van de nieuwste versie, die objectgeoriënteerd is.

"Die object-oriëntatie is natuurlijk belangrijk. Maar persoonlijk vind ik de nieuwe validatiefaciliteiten veel interessanter. Daar kunnen talen als Java en C++ niet aan tippen", aldus Ebbinkhuijsen die aangeeft dat Cobol de laatste veertig jaar een geweldige ontwikkeling heeft doorgemaakt. "Waren de eerste versies voornamelijk gegevensgeoriënteerd, de nieuwste versies zijn veel meer beïnvloed door het procesmatig denken. Dat komt onder meer tot uiting in de validatietesten. Een voorbeeld: wanneer ik een invoerrecord krijg, dan moet die altijd op inhoud worden gecontroleerd. Als daarin een naamveld voorkomt, dan mogen er bijvoorbeeld alleen maar letters, spaties en liggende streepjes in staan. In alle programmeertalen moet je die validatietesten uitschrijven in je programma's. Dat is heel veel werk, en vraagt uiterste zorgvuldigheid van de programmeurs. Bovendien is het onderhoud lastig wanneer de regels wijzigen en aangepast moeten worden. Dat geldt ook voor de huidige Cobol-applicaties. In de nieuwe versie benoemen we de validatieregels bij de veldbeschrijvingen zelf. Het enige wat je nu als programmeur nog hoeft te doen, is de opdracht 'validate' intypen op die plaatsen waar een validatietest moet worden gedaan", aldus Ebbinkhuijsen die zijn carrière in het begin van de jaren zestig als programmeur bij de Staatsmijnen begon.

Daar hadden ze de beschikking over een Univac-computer. In het voorjaar van 1962 volgde Ebbinkhuijsen een Cobol-opleiding. Er waren zes cursisten en negen docenten uit verschillende landen. De opleiding werd gegeven door Grace Hopper. Zij was in 1943 de eerste programmeur voor de beroemde Harvard Mark I, en later bij Univac. Daar ontwikkelde zij in de jaren vijftig een automatisch programmeersysteem, dat zij de A-O compiler noemde. De compiler werkte wel, maar was commercieel geen succes. Het vertalen van de meest

simpele programma's duurde al een uur en de programma's die werden gegenereerd, waren tergend langzaam. Grace Hopper speelde een uitermate actieve rol in het promoten van Cobol. Cobol is erg beïnvloed door de taal Flow-Matic die zij voor haar A-O compiler bij Univac had ontwikkeld. Via onder andere IBM, Philips, Control Data en het Novi kwam Ebbinkhuijsen uiteindelijk in 1977 bij Albert Heijn terecht. Daar heeft hij zich, totdat hij in 2000 met de VUT ging, meestentijds als data-architect bezig gehouden met het ontwerpen van logische databases.

Standaardisatievoorstellen

Schaker en automatiseerder wijlen Max Euwe heeft in de jaren zestig de Nederlandse Cobol Commissie opgericht. Hij vroeg Ebbinkhuijsen in 1966 om lid te worden. Ebbinkhuijsen is nu al weer 28 jaar voorzitter van de club die haar bloeiperiode in het begin van de jaren tachtig beleefde en nu nog zo'n 12 leden telt. "In de jaren zeventig waren er in diverse landen Cobol-commissies waren. Maar de standaarden werden opgesteld door de Amerikaanse commissie, waarop geen enkel ander land invloed had. Eigenlijk een vreemde gang van zaken. Vandaar dat op voorstel van Nederland door de internationale programmeertalencommissie, waarvan ik via het Nederlands Normalisatie Instituut lid was, een internationale toezichthoudende Cobol Experts Group in het leven werd geroepen. Ik herinner me de eerste vergaderingen nog. Die stonden bol van het wantrouwen tussen Amerikanen en Europeanen. Dat is tegenwoordig wel anders. De standaard wordt nu door een internationale commissie vastgesteld, maar het uitvoerende werk wordt nog steeds gedaan door de Amerikaanse commissie", aldus Ebbinkhuijsen, die sinds vier jaar deel uitmaakt van het Amerikaanse gezelschap. "Nergens in de statuten staat dat er alleen Amerikanen lid van mogen worden. Dus toen het werk het toeliet, heb ik mij in 1997 aangemeld als lid. Het is een heel apart gevoel om lid te zijn van een commissie waar ik altijd heel erg tegenop heb gezien. Dat was toch de club waar Cobol werd 'gemaakt'. Wanneer je vroeger als programmeur handboeken bestudeerde, dan wist je dat ze gebaseerd waren op teksten die door deze commissie waren opgesteld. Maar het is daarbij ook heel boeiend en interessant werk. De leden komen zes keer per jaar 1 à 2 weken bij elkaar."

Volgens Ebbinkhuijsen is het een heel secuur werkende groep mensen, die in drie tot vier vergaderingen een standaardisatievoorstel beoordelen. In de eerste vergadering wordt het globale concept besproken. In de tweede -en eventueel derde- bijeenkomst wordt het concept bijgeschaafd. In de laatste vergadering wordt er gestemd. "In de Amerikaanse commissie zitten afgevaardigden van leveranciers en gebruikersorganisaties. Ik vind het heel boeiend om te zien hoe in vergaderingen soms politiek wordt bedreven. Ik herinner me een discussie over de uitbreiding van edit-faciliteiten. Het voorstel was om de 'picture' uit te breiden. Dat is een plaatje waarmee een veldbeschrijving wordt vastgelegd. De vertegenwoordigers van twee leveranciers waren daar fel op tegen. Ik snapte dat niet. Het ging toch maar om het toevoegen van twee woordjes in het standaard document. Later bleek dat de twee leveranciers de huidige situatie hard in hun compilers hadden gecodeerd. Daarmee waren precies 16 bits gemoeid.

Uitbreiding van de edit-faciliteiten met die twee woordjes zou inhouden dat ze van 16 bit naar 24 bit moesten uitbreiden. In deze fase van de levenscyclus hadden die twee geen zin om daarvoor extra investeringen te plegen. Overigens is de tekst wel aangenomen, zodat die twee leveranciers hun compilers toch

zullen moeten aanpassen, wanneer ze tenminste hun producten aan de standaards willen laten voldoen. Andersom zie je ook wel dat bepaalde leveranciers iets nieuws in hun eigen producten inbouwen en vervolgens proberen een dergelijke noviteit als standaard te laten aannemen. Als dat lukt, hebben zij tenminste een voorsprong op de rest. Het zijn soms verwarrende discussies, waarbij officieel anders wordt gesproken dan in de wandelgangen."

Opleving mainframe

Het is het politieke steekspel in combinatie met de zorgvuldigheid die de meeste commissieleden willen betrachten, die de vertraging in het uitbrengen van Cobol 2000 heeft veroorzaakt. Daarnaast eisen de aangepaste internationale procedures ook hun tol. Het zal de mensheid niet echt verontrusten. Voor Cobol dat een soort van eeuwigheidswaarde lijkt te hebben, is een jaar uitstel niet zo erg. Cobol is er en zal er voorlopig nog wel blijven. Er is domweg geen tijd, geld en expertise om al die Cobol-programmatuur in een andere taal en voor een ander hardwareplatform te herschrijven. Bovendien doen mainframes en hun applicaties het op zich goed. Ze werken snel en tegen een redelijke prijs/prestatieverhouding. Dus waarom veranderen? Daar komt nog iets bij: er is een heel grote marktspeler veel aan gelegen om deze situatie lang, heel lang te continueren. En dat is IBM. Big Blue is een van de weinige mainframeleveranciers die alle revoluties heeft doorstaan. Wat heet, de computergigant haalt nog steeds een forse omzet, en dito winst, uit de mainframehandel. IBM heeft een mooi plan ontwikkeld om de mainframewereld als basis te laten fungeren voor moderne intranet- en webapplicaties. In dit scenario is DB2 de belangrijkste database en Cics (Customer Information Control System, oorspronkelijk oltp-software voor het aansturen en beheren van werkstations en pc's die aan het mainframe gekoppeld zijn) de transactiemanager voor webapplicaties. En Cobol de programmeertaal voor de applicaties die tegelijkertijd voorziet in de business logica. Talen als Java en Visual Basic kunnen worden gebruikt voor het maken van de grafische interfaces op de clients die de data verzenden naar en ontvangen van de Cics-omgeving.

Het is, zo stelt Ebbinkhuijsen, een aardige gedachte. Hoewel de ict-wereld wat complexer in elkaar zit dan dertig jaar geleden, toen inderdaad één computerleverancier de wet kon stellen, meent Ebbinkhuijsen dat dit door IBM geschetste scenario best wel eens kans van slagen kan hebben. IBM is immers nog steeds een dominante partij op de mainframemarkt. Maar er zijn nog andere redenen. Na een wild-west periode van tien jaar waarin decentralisatie van de automatisering hoogtij vierde, zien veel ict-managers in dat een zekere mate centralisatie van ict-faciliteiten de beheersbaarheid, en dus het kostenpatroon, in positieve zin beïnvloedt. Vanuit deze optiek is het veel eenvoudiger om een grote en complexe applicatie op basis van één databaseserver te ontwikkelen dan een dergelijke applicatie op verschillende kleine, decentraal opgestelde servers te repliceren. Een revival van het mainframe, en dus van Cobol, zou wel eens in het verschiet kunnen liggen. Temeer omdat Cics als een krachtige en veilige transactieprocessor zich veruit superieur toont aan alle andere transactiesoftware die in een client/server-omgeving zijn ontstaan.

Stoffig imago

Ebbinkhuijsen is echter realist genoeg om te erkennen dat Cobol in de wereld van de pc-netwerken niet snel een dominante rol kan gaan spelen. Pogingen die

in het verleden zijn ondernomen om Cobol-applicaties te converteren van een Unix-midrange omgeving naar pc-netwerken zijn allemaal uitgelopen op een mislukking uitgelopen. "Fujitsu ontwikkelt een Cobol-compiler die in een objectgeoriënteerde client/server-omgeving kan worden ingezet. De ontwikkelaars baseren zich daarbij op de nieuwe Cobol-2000 standaarden. Ze gokken erop dat Cobol wellicht een rol kan gaan spelen als ontwikkelgereedschap in het .Net-framework dat Microsoft heeft aangekondigd. Het komt er dan op neer dat je Cobol-*classes* ofwel objecten kunt bouwen die samenwerken met objecten die met andere programmeertalen zijn ontwikkeld. Dat betekent dat ontwikkelaars in de nabije toekomst die taal kunnen gebruiken die het best past bij een bepaald type object. Bijvoorbeeld Visual Basic voor de *gui-classes* en Cobol voor de *business-classes*. Een boeiend perspectief dat prima zou passen bij het idee dat IBM heeft gelanceerd. Of deze ideeën ooit werkelijkheid worden en een nieuwe bloeiperiode voor Cobol zullen inluiden? Dat is moeilijk te zeggen. Je merkt dat er steeds minder Cobol-programmeurs zijn. De oude generatie verlaat zo langzamerhand de arbeidsmarkt, en aan de onderkant komt er nauwelijks iets bij. Cobol is niet sexy voor de jonge generatie. Misschien dat dat stoffige imago verdwijnt met de komst van Cobol 2000. Feit is en blijft evenwel dat er nog vele jaren lang Cobol-kennis noodzakelijk blijft. Al was het alleen maar om de bestaande applicaties te onderhouden en aan te passen. En om nieuwe applicaties aan de bestaande software te koppelen", meent Ebbinkhuijsen, die druk bezig is met het schrijven van een nieuw Cobol-handboek. In het begin van de jaren zeventig schreef hij het eerste Nederlandstalige leerboek, gebaseerd op Cobol-68. De huidige editie, dat Cobol-85 als onderwerp heeft, heeft een omvang van 800 pagina's. Het boek over Cobol-2000 zal zo'n 1100 bladzijden gaan beslaan. Die extra ruimte is nodig om alle uitbreidingen en veranderingen te beschrijven. En dan is het gedeelte over object-oriëntatie daar niet bij inbegrepen. Want dat wordt een heel nieuw boekwerk.

COK DE ZWART
FREELANCE MEDEWERKER

© VNU Business Publications Nederland

Dit artikel is afkomstig uit het online weekbladarchief van Computable.
Het is te vinden op: <http://www.computable.nl/artikels/archief2/d03ra2br.htm> .

- [Terug](#)