



Maher Awad over objectoriëntatie

**‘Veel van  
wat  
Octopus  
eerder al  
gebruikte  
is in UML  
terechtgekomen’**

## TRAINING voor realtime embedded systemen

Octopus is een objectgeoriënteerde ontwikkelmethode voor realtime embedded systemen. Ze omvat het gehele ontwikkeltraject, van requirements en specificatie tot en met implementatie en testen. De methode is uitgewerkt bij het Nokia Research Center in Finland en in 1996 beschreven door Maher Awad, Juha Kuusela en Jürgen Ziegler. PT Embedded Systems sprak met Maher Awad.

HANS VAN THIEL

**PT Embedded Systems:** 'Wij hoorden dat u op dit ogenblik systeemmanager bent van een *new generation software platform* bij Nokia Telecommunications in Finland. Kunt u daar iets over vertellen?

**Maher Awad:** 'Dit project is op het ogenblik nog geheim maar zal aan het einde van dit jaar naar buiten worden gebracht. Ik kan er nu helaas alleen over zeggen dat het om een softwareplatform gaat, een infrastructuur die de basis zal vormen voor een geheel nieuw product.'

**PT Embedded Systems:** 'Hiervoor was u hoofd van de afdeling software binnen de Internet Access Division van Alcatel Telecom in België, en daarvoor werkte u ook bij Nokia. U werd speciaal bij Alcatel aangetrokken om op die afdeling objectgeoriënteerd ontwerpen in te voeren, maar wanneer bent u zelf overtuigd geraakt van de voordelen van OO voor embedded software?'

**Maher Awad:** 'Al in 1991 werd bij Nokia OO gebruikt voor de 'synchronised digital hierarchy'. Dit SDH omvat een hele softwarelijn waar meer dan 100 ontwikkelaars aan gewerkt hebben en die helemaal in C++ is geschreven. Bij Nokia staat men open voor nieuwe ontwikkelingen en wil men als het even kan state of the art gebruiken. In 1993 is men daarom met een Strategic Software Initiative begonnen en daaruit is de Octopus ontwikkelmethode voortgekomen. In 1996 is die methode dan beschreven in het boek *Object Oriented Technology for*

*Real Time Systems*. Omdat men bij Alcatel geïnteresseerd was geraakt in Octopus als werkwijze voor het hele ontwikkeltraject ben ik gevraagd de methode te introduceren.'

**PT Embedded Systems:** 'Was het moeilijk ontwikkelaars te overtuigen van het nut van verandering?'

**Maher Awad:** 'Er bestaat altijd een aanvankelijke weerstand tegen verandering, maar in Octopus kan bestaande programmatuur goed worden ingepast. Hierdoor kan OO geleidelijk worden ingevoerd en dat maakt de uiteindelijke acceptatie een stuk makkelijker. Software-engineers gaan zelf de voordelen zien.'

**PT Embedded Systems:** 'Bjarne Stroustrup, de schepper van C++, zei in een interview met ons dat er geen enkel programma bestaat in C++ dat groter hoeft te zijn of minder snel dan het gelijksoortige in C. Hij zei ook dat C++ volledig geschikt is voor embedded systemen, zeker nu de ISO/ANSI standaard gereed is en de tools steeds beter zijn geworden. Wat is uw reactie daarop?'

**Maher Awad:** 'Dat kan ik volledig bevestigen, al geloof ik niet dat de standaardisatie hiervoor direct van belang is. Het gaat er eerder om dat de ontwikkelaars beseffen wat de kosten zijn van bepaalde features en weten welke te gebruiken zijn onder welke omstandigheden. Bij Alcatel hebben we voor

de meest kritieke component van het product, de 'asynchronous digital subscriber loop', OO toegepast en dat modemprotocol is in C++ geïmplementeerd. Bij Nokia hebben we met metingen - let wel, dit is niet zomaar een mening - aangetoond dat C++ beter presteerde dan C voor tijdskritieke applicaties.

**PT Embedded Systems:** 'Wat vindt U van Java?'

**Maher Awad (glimlacht):** 'Ik hou van Java. Het is eenvoudiger en daarom beter. Het heeft extra mogelijkheden zoals *threads*, *synchronisation* en *garbage collection*. Het belangrijkste is echter wel dat er zulke uitgebreide componentbibliotheken beschikbaar zijn. Voor embedded systemen kan het een nadeel zijn dat Java trager is en meer overhead heeft. In het geval van kritieke *time-requirements* of geheugenbeperkingen is Java niet goed te gebruiken.'

**PT Embedded Systems:** 'Testen valt ook onder de Octopus ontwikkelmethode?'

**Maher Awad:** 'Het bepalen van de performance is onderdeel van de Octopus ontwerpfasen. Je moet daar niet mee wachten tot je gaat testen, want dan is het al te laat. Testen is natuurlijk wel nodig, maar dit komt na een performance-analyse.'

**PT Embedded Systems:** 'In het boek over Octopus staat dat ontwikkelaars nogal eens de fout maken om al in de designfase met het eigenlijke programmeren te beginnen.'

**Maher Awad:** 'Softwaredesign bestaat niet uit het schrijven van code, maar uit het bepalen van stappen die tot een oplossing leiden. Het gaat in deze fase niet om het programmeren van het probleem, maar om het oplossen ervan.'

**PT Embedded Systems:** 'Octopus plaatst alle specifieke hardwarefuncties in een welomschreven interface, de 'hardware wrapper'.'

**Maher Awad:** 'In de meeste toepassingen is 80 % van de programmatuur hardware-onafhankelijk. In Octopus wordt alles dat specifiek is voor de hardware, bijvoorbeeld interruptrouti-

'Ik zou zelf nooit meer voor een bedrijf willen werken dat geen objectoriëntatie toepast.'

'Er is met cijfers aangetoond dat C++ beter presteert dan C voor tijdskritieke applicaties.'

'Ik hou van Java. Het is eenvoudiger en daarom beter.'

nes en het aansturen van in- en uitvoerpoorten, ondergebracht in een interface; de 'hardware wrapper'. Door deze functies echt in te bedden wordt het hergebruik van de software, de *reusability*, sterk bevorderd.'

**PT Embedded Systems:** 'U zegt ook dat in het ideale geval software en hardware van realtime embedded systemen samen worden ontwikkeld, maar dat in de praktijk de hardware meestal al gekozen is voordat de software-engineers erbij worden gehaald.'

**Maher Awad:** 'De hardware valt ook onder Octopus en er zijn feedbacktechnieken naar de hardware toe. Het kan voorkomen - en dat is in de praktijk ook veelvuldig gebeurd - dat vanuit de software-ontwikkeling blijkt dat hardware-resources kunnen worden aangepast. Het kan bijvoorbeeld tijdens het software-ontwikkelen duidelijk worden dat een vierbits component voldoende is voor een bepaalde functie, in plaats van de aanvankelijk gekozen achtbits versie. In meer fundamentele zin moet je echter denken aan de integratie van asic-chipontwerp in het gehele proces van systeemontwikkeling. Maar in Octopus kan ook, als de hardware echt van tevoren is vastgelegd, uitgegaan worden van daardoor bepaalde randvoorwaarden.'

**PT Embedded Systems:** 'Wat is de rol van een besturingssysteem in dit verband?'

**Maher Awad:** 'Objectstructuren moeten worden afgebeeld op een optimaal aantal besturingssysteem-functies. Het ontwerp wordt niet bepaald door processen maar door objecten. OO-analyse is niet goed mogelijk als je ontwikkeling door processen wordt aangestuurd en daarom gaat Octopus uit van *implicit concurrency* in de analysefase. De objecten worden verondersteld concurrency zelf af te kunnen handelen. In de ontwerpfase wordt concurrency dan expliciet en Octopus gebruikt hier een stelsel van prioriteiten waarmee alle *time constraints* duidelijk en in onderlinge samenhang te beschrijven zijn.'

**PT Embedded Systems:** 'Welke relatie heeft Octopus met de 'unified modeling language'?'

**Maher Awad:** 'Octopus gebruikt de UML-notatie. De 'unified modeling language' is, zoals de naam eigenlijk al



## TRAINING OCTOPUS

Maher Awad zal van 4 tot en met 8 oktober een vijfdaagse training verzorgen over de Octopus-methode bij Bergson Software Tools in Eindhoven. Meer informatie op 040 294 63 33 en [www.bergson-tools.com](http://www.bergson-tools.com).

zegt, geen methode maar een taal met een eigen grammatica. In ons boek wordt UML niet genoemd omdat UML pas later is gedefinieerd. Maar wij maakten gebruik van representaties, denk bijvoorbeeld aan Harel statecharts, die later ook in UML zijn opgenomen. Overigens heeft Octopus ook enige specifieke notaties, maar het is in grote lijnen UML-compatible. In ons boek uit 1996 noemen we OMT en Fusion, maar we hebben meer dan twintig methoden bestudeerd en daaruit opgenomen wat we goed vonden en vinden. De ontwikkeling van een methode is natuurlijk een continu proces. Het is leuk dat veel van wat Octopus eerder al gebruikte ook in UML is terechtgekomen.

**PT Embedded Systems:** 'Is Octopus moeilijk te leren?'

**Maher Awad:** 'Octopus is veel technischer dan andere methoden en het omvat bovendien het totale ontwikkeltraject in al haar aspecten. Ik zou zeggen dat het moeilijk is omdat het zo'n breed bereik heeft. Daar staat tegenover dat het incrementeel is en geleidelijk ingevoerd kan worden. Het is ook schaalbaar naar de grootte van het project en het is niet beperkt tot realtime systemen alleen. Octopus wordt momenteel overal bij Nokia gebruikt en er komt steeds meer belangstelling uit andere hoeken.'

**PT Embedded Systems:** 'Zijn er CASE-tools voor Octopus die de ontwikkelaar kunnen helpen?'

**Maher Awad:** 'Zoals ik al zei gebruikt Octopus UML en daarom kan worden ontwikkeld met elke tool die UML ondersteunt. Er zijn ook tools die speciaal

op Octopus zijn toegesneden, zoals Paradigm Plus.'

**PT Embedded Systems:** 'Waarom is Octopus minder geschikt voor het ontwikkelen van wat u noemt *hard-critical real-time systems*?'

**Maher Awad:** 'Bij *hard-critical* moet u denken aan systemen waarvoor onder geen enkele omstandigheid ooit een *miss* is geoorloofd. Bij *soft-critical* is misschien een misser eens per jaar niet zo erg. Systemen die absoluut nooit mogen weigeren zijn duidelijk anders, al biedt Octopus ook daarvoor ondersteuning met *advanced event analysis*.'

**PT Embedded Systems:** 'Eventuele conflicten worden zichtbaar door objecten die elkaar aanroepen te groeperen?'

**Maher Awad:** 'Ja, als alle objecten bepaald zijn, worden ze gegroepeerd aan de hand van hun verbanden. Objecten die tot meer dan één groep behoren, en dat is meteen uit de modellen af te leiden, kunnen conflicten geven. Voor deze gevallen moeten dan prioriteiten vastgesteld worden. Hier wordt ook de concurrency expliciet. Er moet niet alleen worden gekeken naar hoe processen kunnen worden afgebroken om een ander proces mogelijk te maken, maar ook naar data-integriteit. Het fundamentele idee van een object is dat gegevens alleen via de eigen methoden bereikbaar zijn, en als je een proces onderbreekt voor een ander kan in principe het tweede proces data veranderen waar het eerste nog mee bezig is. Dit moet uiteraard goed begrepen en behandeld worden. Verder zijn er behalve synchrone misschien allerlei soorten asynchrone processen en dat komt allemaal aan de orde in deze ontwerp-fase.'

**PT Embedded Systems:** 'Ziet u een toekomst weggelegd voor formele methoden, met name correctheidsbewijzen van embedded software?'

**Maher Awad:** 'Wij maken er geen gebruik van. Het schrijven van formele specificaties in een taal als Z introduceert nogal wat overhead. Dat wil niet zeggen dat formele technieken niet van belang kunnen zijn. Misschien is 1 à 2 procent van de programmatuur zo kritisch dat een mathematisch correctheidsbewijs noodzakelijk wordt. Dan moet je het doen.'

**PT Embedded Systems:** 'Worden er, om daar nog eens op terug te komen, speciale eisen gesteld aan een realtime besturingssysteem?'

**Maher Awad:** 'Octopus is onafhankelijk van tools, programmeertalen en besturingssystemen. Het veronderstelt een *basic preemptive os* dat alle *concurrency-levels*, *multitasking*, *message-passing* en *synchronisation* ondersteunt.'

**PT Embedded Systems:** 'Windows CE?'

**Maher Awad:** 'Maakt het makkelijker door je een Windows-interface te geven. Nadeel is dat je aan Microsoft vastzit en aan COM en DCOM, terwijl CORBA toch echt de standaard is. Windows CE heeft ook een grotere *footprint*. Het is slechts mijn eigen mening, maar voor tijdkritische systemen vertrouw ik het niet. Embedded systemen mogen niet crashen. Nokia gebruikt het Symbian OS, dat samen met Ericson, Motorola en anderen is ontwikkeld voor de nieuwe generatie mobiele telefoons. Dit heeft een kleinere voetafdruk, het is een open besturingssysteem en, niet onbelangrijk, vrij van royalties.'

**PT Embedded Systems:** 'Misschien kunnen we afsluiten met een algemene vraag? Wat vindt u van OO voor embedded systemen? Is dat de toekomst voor alle applicaties?'

**Maher Awad:** 'Laat ik het zo zeggen: ikzelf zou nooit meer voor een bedrijf willen werken dat geen objectoriëntatie toepast.'

**PT Embedded Systems:** 'Ten slotte een vraag uit pure nieuwsgierigheid. Uw naam Maher Awad is een ongebruikelijke hier, maar u komt oorspronkelijk uit Palestina?'

**Maher Awad.** (*glimlachend*): 'Ik heb zowel een Palestijns, een Jordaans als een Fins paspoort, maar ik ben geboren in een plaatsje niet ver van Jeruzalem, en heb onderwijs gevolgd in Palestina en Jordanië. Behalve software-engineering heb ik ook computer-engineering en electrical engineering gedaan. Na mijn opleiding ben ik bij Nokia gaan werken. Vanuit die functie ben ik betrokken geraakt bij OO voor realtime systemen. □

'Het schrijven van formele specificaties in een taal als Z introduceert nogal wat overhead. Dat wil niet zeggen dat formele technieken niet van belang kunnen zijn. Maar wij gebruiken ze niet.'

'In de meeste toepassingen is 80 % van de programmatuur hardware-onafhankelijk.'