



Interview met UML-grondlegger Ivar Jacobson

# ‘**Waarom zou je UML executeer**

Ivar Jacobson is één van de grondleggers van UML. Dit jaar moet de nieuwe versie van de standaard uitkomen, namelijk UML 2.0. 'Dat betekent vooral een verduidelijking en vereenvoudiging van de UML-specificatie zelf. Maar het is nog niet helemaal bekend hoe het eindresultaat zal uitvallen, omdat er twee voorstellen bij de OMG liggen.' Verder verwacht Jacobson dat code en modellen steeds nauwer worden geïntegreerd. 'Waarom zou je nog onderscheid maken?'

HANS VAN THIEL

**PT Embedded Systems:** Dat u samen met Grady Booch en James Rumbaugh tot de grondleggers van UML (unified modeling language) behoort is genoegzaam bekend. U bent alle drie bij Rational gaan werken, maar UML is een openbare standaard van de Object Management Group (OMG). Nu is UML niet voor de volle honderd procent van Rational afkomstig, maar wel voor een groot deel. Hoe is de standaardisatie bevallen? Rational had het ook voor zich zelf kunnen houden.

**Ivar Jacobson:** 'De standaardisatie is zeer positief, een taal hoort niet van een bedrijf te zijn. Bijna een jaar voor ik bij Rational kwam sprak ik al met de OMG. Grady sloot zich hierbij aan en al vrij snel ging Richard Soley van de OMG akkoord. Bedrijven worden overgenomen of fuseren en een algemene taal als UML hoort daar niet door te worden beïnvloed.'

**PT Embedded Systems:** UML is niet gebonden aan een ontwerpmethode.

**Ivar Jacobson:** 'Het ontwerp van een taal is niet onafhankelijk van de manier waarop die gebruikt wordt. UML is procesonafhankelijk maar je hebt toch een werkwijze nodig. Het Objectory-proces

(Jacobson's methodologie vóór UML, red.) is het Rational Unified Process (RUP) geworden. RUP is eigenlijk een enorme knowledge base die wel twintig tot veertig boeken beslaat over allerlei onderwerpen zoals requirements, business engineering, testen, metriek, noem maar op. Maar het kenmerk is dat al die boeken dezelfde begrippen en terminologie gebruiken en naadloos in elkaar passen. En er zijn overal tools voor.'

**PT Embedded Systems:** Bran Selic (zie PT Embedded Systems, 2001/3, red.) is zeer betrokken geweest bij de standaardisatie van realtime UML. Wordt die standaard inmiddels door tools ondersteund?

**Ivar Jacobson:** 'RUP heeft een speciale sectie voor *realtime* met Rose-RT ondersteuning. RT-UML wordt nu al geïmplementeerd. Het gebeurt op dit moment.'

**PT Embedded Systems:** Dit jaar moet de OMG standaard UML 2.0 uitkomen. Is al bekend wanneer precies?

■ 'Eens zal er een modelleringstaal zijn die zo algemeen is dat die kan worden gebruikt op de basisschool.'

**Ivar Jacobson:** 'Dat wordt een belangrijke revisie maar een deel daarvan, de infrastructuur, zal weinig invloed hebben op eindgebruikers. Die aanpassing betekent vooral een verduidelijking en vereenvoudiging van de UML-specificatie zelf. Maar het is nog niet helemaal bekend hoe het eindresultaat zal uitvallen, omdat er twee voorstellen bij de OMG liggen. Wij hopen en verwachten uiteraard dat ons voorstel het zal worden maar misschien komen er aanpassingen. Ook in 1997, bij de eerste standaardisatie van UML, is het zo gegaan.'

**PT Embedded Systems:** Op de 2001 UML World Conference in New York

hebt u vier, wat u noemt 'macrotrends' aangewezen in software-ontwikkeling. De trends zijn: software-ontwikkeling met herbruikbare componenten, integratie van verificatie in het ontwikkelingsproces, programmeren door eindgebruikers met behulp van 'agents' en het gebruik van UML in alle lagen van computersoftware. Zullen we beginnen met componenten?

**Ivar Jacobson:** 'In banken, verzekeringen en bij luchtvaartmaatschappijen kom je honderden features tegen die steeds maar weer opnieuw worden geïmplementeerd. De oplossing ligt in grote herbruikbare componenten die worden geproduceerd in component-fabrieken en kunnen worden gekocht op een nieuwe-componenten-markt. Het probleem is dat je in zo'n markt moet kunnen weten wie wat levert en wat je dan precies krijgt. Dan moet zo'n component behalve code ook een architectuur-model bevatten, testmodellen waaronder regressie-testen, en in de eerste plaats duidelijk uitgewerkte

*use cases*. Rational heeft het initiatief genomen met een voorstel voor een *reusable asset specification* (RAS). Dat voorstel, dat behalve door ons ook door IBM, Microsoft en nog anderen wordt ondersteund, zal te zijner tijd aan de OMG zal worden aangeboden. RAS zal, zodra de UML 2.0 standaard er is, daar bij aansluiten.'

**PT Embedded Systems:** Zullen herbruikbare componenten moeten worden gecertificeerd? De OMG denkt momenteel aan de invoering van certificering en kwalificatie-procedures voor standaarden.

**Ivar Jacobson:** 'Certificering heeft voor en nadelen, die ik allebei ook wel zie. Als het door consultants moet worden uitgevoerd dan ben ik er geen voorstander van. Er bestaat voor UML-compliance van tools ook geen certificering. Zo lang we die nog niet hebben ...'

**PT Embedded Systems:** U verwacht dat de rol van UML in de toekomst nog veel belangrijker zal worden.

**Ivar Jacobson:** 'UML is geschikt voor het

# niet direct baar maken?

**'Als modellen en code steeds nauwer worden geïntegreerd, waarom zou je dan nog onderscheid maken? Ik zie dit echt gebeuren in de komende tien jaar.'**



modellieren van systemen in allerlei disciplines, niet alleen maar software. Van deze uitbreiding van UML naar boven, kom je via use cases naar ontwerp en via ontwerp naar implementatie weer beneden uit. Je ziet nu al dat 80 - 90% op dat niveau uit Java bestaat en je vraagt je dan af waarom je Java eigenlijk nog nodig hebt. Waarom zou je UML niet direct executeerbaar maken of misschien toepassen als een familie van executeerbare talen? Als modellen en code steeds nauwer geïntegreerd worden, waarom zou je dan nog onderscheid maken? Ik zie dit echt gebeuren in de komende tien jaar.'

**PT Embedded Systems:** Dertig jaar geleden had je 4GL (fourth generation language) en zelfs 5GL dat programmeren overbodig zou maken.

**Ivar Jacobson:** 'Programmeren door leken (Eng. layman programming) is tot nu toe beperkt gebleven tot vastomlijnde gebieden in software maar wat ik zie gebeuren is dat zakenmensen hun eigen software gaan schrijven. De manier om dat mogelijk te maken ligt in het gebruik van intelligente agents. Denk aan een *knowledge base* waar je redeneringen op kunt toepassen en misschien wel honderden agents met kennis over software die kunnen helpen bij de ontwikkeling. RUP zoals dat zich zal ontwikkelen, met tools die RUP gebruiken en omgekeerd, zal zo'n basis kunnen vormen. Denk verder aan agents die patterns kunnen identificeren, agents die deskundig zijn over herbruikbare componenten (Eng. reusable assets), agents die use cases kunnen helpen creëren. Inderdaad, dat zijn een soort wizards maar gebaseerd op regels en met veel meer mogelijkheden. Ik denk dat het over 25 jaar zo ver zal kunnen zijn. Vijfentwintig jaar geleden was de situatie heel anders. Met 4GL zat je in een tunnel waar je niet uit kon komen, nu zit je niet in een tunnel.'

**PT Embedded Systems:** UML is misschien ook geschikt voor hardware-ontwerp en embedded ontwikkelaars zouden heel gelukkig zijn met hw/sw-codesign. Ziet u daar mogelijkheden?

**Ivar Jacobson:** 'Bij Ericsson deden we al in 1975 aan hardware-ontwikkeling met

volgordediagrammen, toestandsdiagrammen, samenwerkingsdiagrammen, enzovoorts. Ericsson zou kunnen beweren dat zij de eerste waren die UML gebruikten, al heette dat toen niet zo. UML is geen nieuwe schepping geweest; het is een integratie van al langer bestaande hulpmiddelen. UML is zo generiek dat het ook heel goed met een hardware-beschrijvingstaal als VHDL is te gebruiken. Codesign deden we ook al heel vroeg. We implementeerden componenten in hardware, software of data. Op een hoger niveau maakt het niet uit of je een digitaal signaal of een software-boodschap beschrijft. In UML is dat vervolgens heel makkelijk aan te geven door een stereotype waarmee je de context volkomen duidelijk maakt. Er ligt op dit moment overigens ook een UML-versie voor *systems engineering* bij de OMG.'

**PT Embedded Systems:** Modelleren is een heel algemeen bruikbare techniek.

**Ivar Jacobson:** 'Zeker in samenwerking met intelligente agents. Kijk, een heleboel jaren geleden droomde ik ervan dat de benadering die ik bij Ericsson ontwikkelde niet belachelijk gemaakt zou worden. Het heeft tien jaar geduurd voordat die werd geaccepteerd. Nu heb ik weer een droom. Eens zal er een modelleringstaal zijn die zo algemeen is dat die kan worden gebruikt op de basisschool. Jonge kinderen zullen leren denken in modellen en als ze ouder worden zullen ze andere modellen gaan gebruiken. Natuurkunde, wiskunde, noem maar op, het zal allemaal onderwezen worden met behulp van die ene modelleringstaal. Het is het oude idee van systeemdenken en we zullen *patterns* ontdekken waarmee dat werkt.'

**PT Embedded Systems:** De vierde trend die u aanwijst is de integratie van softwareverificatie met ontwikkeling. In XP (eXtreme Programming) wordt de test geschreven nog voordat de functie is geïmplementeerd. Bedoelt u iets dergelijks en wat vindt u in dat verband van Agile Modeling?

**Ivar Jacobson:** 'In mijn opinie is dat allemaal toch vooral boekenwijsheid. De validatie ligt bij ons in de use cases die

zowel specificatie, ontwerp als testen aansturen. De use case is gelijk aan de test case. Waar ik Agile Modeling wel de eer voor wil geven is dat zij de vinger hebben gelegd op het belang van snelheid. Maar waar ben je zonder proces, en zonder tools? Dan ga je dertig jaar terug in de tijd!'

**PT Embedded Systems:** Maar er is geen enkele tool die UML voor de volle honderd procent ondersteunt.

**Ivar Jacobson:** 'Dat zal door de toename van het aantal gebruikers steeds beter worden.'

**PT Embedded Systems:** Cris Kobryn van Telelogic, die betrokken is bij de OMG standaardisatie, zegt dat SDL (Specification and Description Language) een sterke invloed heeft gehad op UML 2.0.

**Ivar Jacobson:** 'SDL stamt uit de jaren tachtig, heeft inmiddels een aantal ontwikkelingen doorgemaakt, maar is nooit echt op grote schaal doorgebroken. Maar het had toch een aantal belangrijke bijdragen te leveren die we aanvankelijk niet in aanmerking hebben genomen, en die nu voor UML 2.0 wel worden overwogen. SDL werd trouwens heel veel gebruikt door Ericsson, zo sterk zelfs dat het ook wel de *Ericsson language* werd genoemd.'

**PT Embedded Systems:** Hebt u lang bij Ericsson gewerkt?

**Ivar Jacobson** (breed grijnzend): 'Ik heb een gouden horloge van ze gehad. Ik werkte bij Ericsson, later bij mijn eigen bedrijf Objectory, maar dat werd door Ericsson overgenomen. Ja (met een twinkeling in de ogen), je komt er niet zo gemakkelijk vanaf. Maar de jaartallen kloppen niet, want ik heb voor meerdere bedrijven tegelijk gewerkt, op een gegeven ogenblik zelfs voor zowel Ericsson, Objectory als Rational. Bij Rational zit ik binnenkort vijftien jaar en dan krijg ik een leren jack. Je krijgt elke vijf jaar wat, maar na vijftien jaar is dat een leren jas.'

**PT Embedded Systems:** Wat voor jack?

**Ivar Jacobson:** 'Een pilotenjack. De oprichters van Rational komen oorspronkelijk van de *Air Force Academy*. Grady Booch ook trouwens, maar dat heeft er verder niets mee te maken.'