

Monique Snoeck van de Katholieke Universiteit Leuven:

# “Modellerings technieken zijn nog niet in staat conflicten tussen business-rules te signaleren”

*De belangstelling voor het scheiden van business-rules van de feitelijke applicatie is de afgelopen tijd sterk toegenomen. Door regels niet langer 'hard coded' op te nemen in de toepassing maar in een aparte business process management-laag onder te brengen, is de applicatie veel sneller en gemakkelijker aan veranderende omstandigheden aan te passen. Maar hoe doen we dat nu precies? Door te kiezen voor een volledige scheiding tussen programmacode en BPM-laag? Of gebruiken we liever - als een soort hybride oplossing - een speciale codegenerator waarbij na iedere aanpassing van een business-rule in de programmacode de gehele applicatie snel opnieuw kan worden gecompileerd? Al dit soort onderwerpen vormen het onderzoeksdomein van Monique Snoeck, hoogleraar aan de Katholieke Universiteit Leuven.*

“Mijn belangstelling gaat vooral uit naar business modelling en dan met name naar business modelling die gericht is op het ontwikkelen van informatiesystemen. Het draait dus allemaal om domeinkennis en domeinmodelling. Voorheen gebeurde domeinmodelling vooral op een statische manier. We kijken dan naar de objecten die in het domein een rol spelen en onderzoeken vervolgens hun onderlinge relatie. Er bestaat echter ook zoiets als dynamische domeinmodelling. Wat gebeurt er nu precies wanneer in een systeem bijvoorbeeld een 'klant' wordt aan-

gemaakt of een 'order'? We noemen dat ook wel de gedragskant van business- of domeinmodelling.”

Aan het woord is Monique Snoeck. Sinds 1995 is zij als hoogleraar verbonden aan de vakgroep Beleidsinformatica van de faculteit Economische en Toegepaste Economische Wetenschappen van de Katholieke Universiteit Leuven. “De KU Leuven kent drie faculteiten die zich met informatica bezighouden. Bij de faculteit Toegepaste Wetenschappen en de faculteit Wetenschappen staat vooral het ontwikkelen van de infor-

maticatechnologie centraal, zoals het ontwikkelen van operating systems, databasemanagementsystemen of een stukje embedded software in een GSM-telefoon. Bij de faculteit Economische en Toegepaste Economische Wetenschappen gaat het vooral over de vraag: hoe gebruik je deze technieken nu precies in een bedrijfscontext?”

## Domme tools

Na haar studie informatica trad Snoeck in dienst bij een Belgische bank. “Ik heb daar gewerkt op een afdeling die zich bezighoudt met methodieken en technieken. Eén van mijn taken was het doen van onderzoek naar casetools, zodat de bank een goed beargumenteerde keuze voor een aan te schaffen hulpmiddel kon doen. Wat mij daarbij opviel was dat casetools zo - ik zou haast zeggen - 'dom' waren. Dat moet toch beter kunnen, dacht ik zelf. Die gedachte vormt de rode draad door mijn onderzoek hier aan de universiteit.”

Inmiddels zijn we bijna tien jaar verder. Zijn casetools al wat slimmer geworden? “Nauwelijks”, beklamt Snoeck. “Of misschien kan ik beter zeggen: een beetje. De mogelijkheden die codegeneratoren bieden zijn bijvoorbeeld wel verbeterd. Wat mij echter vooral opvalt is dat de 'awareness' onder informatici dat het anders en beter moet duidelijk is toegenomen. Dat is ook logisch. Wij bouwen

tegenwoordig informatiesystemen die vele malen complexer zijn dan pakweg tien of vijftien jaar geleden. De tools die we daarbij gebruiken, groeien weliswaar in functionaliteit, maar dat is nog niet voldoende. Eigenlijk zouden tools beduidend sneller in mogelijkheden moeten groeien dan tot nu toe het geval is, willen we tenminste meer grip krijgen op het gehele domein van de applicatieontwikkeling.”

Daarmee hebben we een weerbarstig onderwerp bij de kop. Eigenlijk al sinds de eerste dagen van de geautomatiseerde informatieverwerking hebben informatici grote moeite met het vastleggen van de specifieke eisen en wensen van de gebruiker. Hoewel er de afgelopen jaren belangrijke stappen vooruit zijn gezet, gaat het nog altijd met enige regelmaat fout. Ook het denken in business-rules verandert daar op zich weinig aan.

### Flink doorpraten

Bij BPM gaat de meeste aandacht naar de volgorde van de stappen: in welke volgorde vinden de stappen in een proces nu precies plaats? “Neem dit voorbeeld dat ik zelf in de praktijk heb meegemaakt bij een bedrijf dat apparatuur voor internettoegang levert. Stel: een klant koopt een product. Formeel wordt dan een uit drie stappen opgebouwd proces gestart. In het informatiesysteem dat dit proces ondersteunt, wordt eerst een klant aangemaakt. Vervolgens wordt de order ingevoerd, die daarna wordt doorgestuurd naar de serviceafdeling voor het uitvoeren van de opdracht. De regel lijkt hier dus op het eerste gezicht: eerst de klant aanmaken, dan de order invoeren en dan de acceptatie door de serviceafdeling.”

“Pas na flink doorpraten werd duidelijk dat er echter wel degelijk in bepaalde situaties uitzonderingen op



*“Het is verstandig om business-rules uit de applicatie te tillen en onder te brengen in een aparte business process management-laag. Daardoor zal de flexibiliteit van de applicatie aanzienlijk toenemen.”*

deze regel bestonden. Soms wilden verkopers namelijk bij wat ingewikkelde gevallen eerst van de serviceafdeling weten of de aan de klant aangeboden oplossing technisch eigenlijk wel mogelijk was. De sequentie van de stappen is daarmee dus in sommige gevallen anders, namelijk: klant aanmaken, acceptatie door serviceafdeling, dan pas inboeken van de order en vervolgens uitvoeren van de opdracht. In zo’n geval kun je dus niet spreken van een business-rule, maar is sprake van een business-policy.”

### Wees voorzichtig

Het verschil tussen deze twee termen zit voor Snoeck in het verplichtende karakter. Een business-rule is een

verplichting, terwijl bij een business-policy afwijkingen mogelijk zijn. Dat is een belangrijk onderscheid, vindt Snoeck. Maar wat is daar bij het ontwikkelen van applicaties de consequentie van? “Het nut van dit onderscheid zit ‘m in het feit dat we inmiddels heel goed weten dat we een business-policy niet ‘hard coded’ in de software moeten opnemen. Doe je dat wel, dan krijg je situaties die we allemaal als consument of klant wel eens hebben meegemaakt. Dan moet zo’n verkoper tegen zijn klant zeggen: ‘De computer weigert dat!’”

Bovendien levert het in de software opnemen van een business-policy vaak overbodig werk op. In de hiervoor genoemde situatie zou het bij-



voorbeeld kunnen betekenen dat de verkoper de oplossing waar twijfels over bestaan eerst als order moet inboeken, deze doorstuurt naar de serviceafdeling, waarna het kan gebeuren dat deze afdeling de order als technisch niet mogelijk weigert. Dan zal die order dus geannuleerd moeten worden, contact met de klant moeten worden opgenomen en dergelijke.

“Wees dus voorzichtig met business-rules. Het is namelijk ook nog eens zo dat business-rules regelmatig gewijzigd worden of gewijzigd moeten kunnen worden. Het is daarom verstandig om business-rules uit de applicatie te tillen en onder te brengen in een aparte business process management-laag. Daardoor zal de flexibiliteit van de applicatie aanzienlijk toenemen. Michael Jackson stelt in zijn Jackson System Development (JSD, red.) methode voor applicatie-ontwikkeling dat de gehele wereld correct in de applicatiestructuur gereflecteerd dient te zijn. Maar dat is dus niet voldoende. De wereld verandert en dus zal ook de applicatie gewijzigd moeten kunnen worden. Het is dus van groot belang dat de juiste dingen in de juiste laag terechtkomen. Aan

de KU Leuven hebben we dat in het verleden mogelijk gemaakt met de zogeheten Merode-ontwikkelmethode. Dat was echter een aanpak die bedoeld is voor gebruik bij maatwerksoftware. Daarom werken we nu aan een aanpak die juist ook bij het gebruik van standaardpakketten kan worden toegepast.”

### Eenheden van coördinatie

“Veel oplossingen voor applicatie-integratie doen dit via een aanpak waarbij één partij of informatiesysteem vraagt en een andere partij de gewenste dienst levert. ‘One way messaging’ dus. Dat is een belangrijk nadeel, want hierbij komen we nooit tot Jacksons reflectie van de reële wereld, want daarvoor hebben we natuurlijk tweeweg- en meer-partijencommunicatie tussen applicaties nodig. Daarom zoeken wij naar wat ik maar ‘eenheden van coördinatie’ zal noemen. Dat wil zeggen dat wij voor elke bedrijfsgebeurtenis vaststellen welke objecten bij dit ‘event’ betrokken zijn, welke voorwaarden de partijen stellen bij het toelaten van de gebeurtenis (de zogenaamde validatieregels, red.) en wat precies het effect is van de gebeurtenis, zoals creatie en wijziging. Deze informatie plaatsen we in een zogeheten ‘object-eventtabel’. Hierin zien we dus precies welke objecten op welke manier bij een bepaalde gebeurtenis zijn betrokken.”

Deze manier van werken biedt nog een voordeel. “Een bedrijfsgebeurtenis is immers een uniek aanspreekpunt voor het manipuleren van een bedrijfsobject. In plaats van berichten naar bedrijfsobjecten te sturen, kan een informatieservice nu het uitvoeren van een gebeurtenis activeren. Dit is met name interessant als dezelfde functionaliteit in verschillende vormen wordt aangeboden. Denk aan het afhalen van geld via bijvoorbeeld een geldautomaat, op het

postkantoor of via het laden van een chipper. Nu kennen deze allemaal hun eigen reeksen van softwarefuncties. Bij een eventgerichte aanpak kunnen al deze diensten echter gebruikmaken van die ene bedrijfsgebeurtenis ‘afhalen’.”

Het geheel aan validatie- en effectregels kan via een op events gebaseerde rule-engine worden gerealiseerd. Maar er zijn ook andere mogelijkheden om business-rules snel aan te passen. “Kijk naar de speciale code-generatoren die we op de markt zien. Daarmee kunnen business-rules toch in de programmeercode zelf worden opgenomen. Deze generatoren maken het mogelijk om de rule aan te passen en de betrokken applicatie vervolgens snel te hercompileren.”

### Niet eenvoudig

Werken met rules is niet eenvoudig, meent Snoeck. “We moeten namelijk zorgen voor twee vormen van consistentie: intern en extern. Met intern bedoel ik: spreken rules elkaar niet tegen? Externe consistentie betekent: begrijp ik wat de gebruikers nu precies vragen? Dit laatste is erg moeilijk te verifiëren en kan alleen worden bereikt via communicatie en - zeg maar - onderhandelen door een analist. Wat die interne consistentie betreft hebben we een probleem en dat is voor mij een interessante onderzoeksvraag: modelleertechnieken zijn namelijk nog altijd niet in staat alle mogelijke conflicten tussen rules te signaleren, laat staan op te lossen.”



**Robbert Hoefnagel**

*Robbert Hoefnagel is freelance journalist.*