

## Integratie van systeem- en procesmodellen

# De brug tussen systeemontwerp en procesmodellering

In projecten waarbij een informatiesysteem moet worden gekoppeld met een workflowapplicatie of een casemanagementtool zijn naast 'traditionele' systeemontwerpen ook procesontwerpen nodig. Het spreekt voor zich dat deze twee ontwerpen goed op elkaar moeten aansluiten. Toch blijkt uit de praktijk dat dit vaak niet het geval is. Het aan elkaar verbinden van modelleertechnieken uit de systeem- en procesmodelleringswereld kan dan soelaas bieden.

Over de integratie van twee werelden en hoe tools daarbij kunnen helpen.

ICT-projecten hebben in het algemeen als doel een systeem te ontwikkelen ter ondersteuning van bedrijfsprocessen. In dit soort projecten vervullen mensen diverse rollen, waaronder die van procesontwerper en systeemontwerper. De producten die zulke proces- en systeemontwerpers opleveren sluiten vaak niet goed op elkaar aan, alhoewel de raakvlakken evident zijn. De oorzaak voor dit probleem ligt voornamelijk op het vlak van communicatie en slechte aansluiting van methoden en technieken die beide groepen gebruiken. Met het op elkaar aansluiten van modelleertechnieken is echter al een stap in de goede richting te zetten.

### Systeem- en procesontwerp

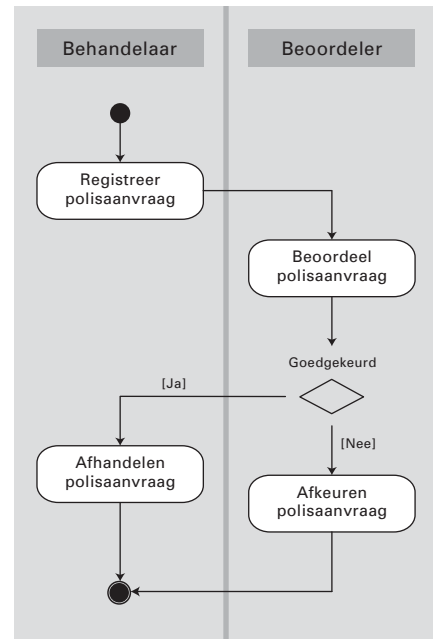
Een in de wereld van het systeemontwerp veelgebruikte ontwikkelmethodiek is UML<sup>1</sup>. De techniek van UML om requirements vast te leggen voor het ontwikkelen van een systeem is het opstellen van use-cases. Deze use-cases worden gedurende het ont-

wikkeltraject verder uitgewerkt in andere UML-modellen (zoals classen en sequence-diagrams) op basis waarvan men code schrijft.

In de wereld van het procesontwerp legt men de requirements vast in proces- of workflowontwerpen die bepaalde processen beschrijven. In zo'n proces zijn meerdere activiteiten opgenomen die het in de use-cases beschreven systeem geheel of gedeeltelijk ondersteunt.

### Integratie van twee werelden

Procesontwerpers en systeemontwerpers hebben vaak hun eigen tool, eigen modelleerconventies en afbakeningscriteria, waardoor de geproduceerde modellen niet goed op elkaar aansluiten. Integratie is mogelijk door de raakvlakken van de verschillende methoden te analyseren en waar nodig aan te passen. Hierna volgt een voorbeeld van de wijze waarop een dergelijk traject kan worden aangepakt. Hierbij worden UML



Afbeelding 1. Een activity-diagram.

en de Ordina procesmodelleertechniek PAM geïntegreerd.

### Procesontwerpen in UML

Een procesontwerp bestaat uit procedures met daarin activiteiten. Een procedure beschrijft de afhandeling van een gebeurtenis (bijvoorbeeld een polisaanvraag). Aan het eind van de procedure is de gebeurtenis volledig afgehandeld. Een activiteit is voor de gebruiker een logische eenheid van werk die hij uitvoert (bijvoorbeeld registratie of beoordeling van de aanvraag). ICT kan een gebruiker bij deze activiteit ondersteunen, met andere woorden: aan een activiteit kan een systeemontwerp worden gekoppeld. Deze procesontwerpen zijn in een modelleertechniek van UML te modelleren door

middel van een activity-diagram (zie afbeelding 1). Hierdoor is het makkelijker een koppeling te maken naar de systeemontwerpmodellen.

#### *Systeemontwerpen aansluiten op het procesontwerp*

UML kent use-cases, die zijn afgebakend op het bereiken van een doel. Om aan te sluiten op het procesontwerp moet de organisatie op dit punt een striktere afbakening op een use-case toepassen, namelijk het afbakeningscriterium van een activiteit. Het doel van de use-case moet het ondersteunen worden van een activiteit uit het procesontwerp. Met andere woorden: voor elke activiteit wordt een use-case geschreven. Het activity-diagram laat de samenhang tussen de activiteiten en de use-cases zien in een logistische vorm.

#### *Het resultaat*

In een project begint men met het bepalen of aanpassen van de procesontwerpen in activity-diagrams. Per activiteit in een procesontwerp moet worden aangegeven of een ICT-systeem deze activiteiten moet ondersteunen. Voor de activiteiten waarvoor ICT-ondersteuning noodzakelijk is kunnen use-cases worden

geschreven op basis waarvan het ontwikkeltraject voor het ICT-systeem kan plaatsvinden. Op deze wijze is er één geïntegreerde wijze van ontwikkelen voor procesontwerp en systeemontwerp, waardoor de aansluiting op elkaar sterk zal verbeteren.

#### **Toolondersteuning**

Een voorbeeld van een applicatie ter ondersteuning van objectgeoriënteerde analyse en ontwerp van een systeem is Rational Rose. Dit tool gebruikt UML-begrippen als basis. Het voordeel van het gebruik van een dergelijk tool is dat volledig 'traceable' specificaties van analyse tot en met ontwerp zijn op te stellen. De toekomstige beheerorganisatie kan dan onder andere makkelijk terugvinden waar bepaalde stukken functionaliteit worden hergebruikt. Daarnaast is het mogelijk om bepaalde performancevoorspellingen te doen, omdat het ontwerp vanaf proces tot en met service traceable is.

#### *Activity-diagrams*

Procesontwerpen worden gemodelleerd in activity-diagrams (zie afbeelding 1). Het activity-diagram in Rational Rose heeft standaardmodelleer-

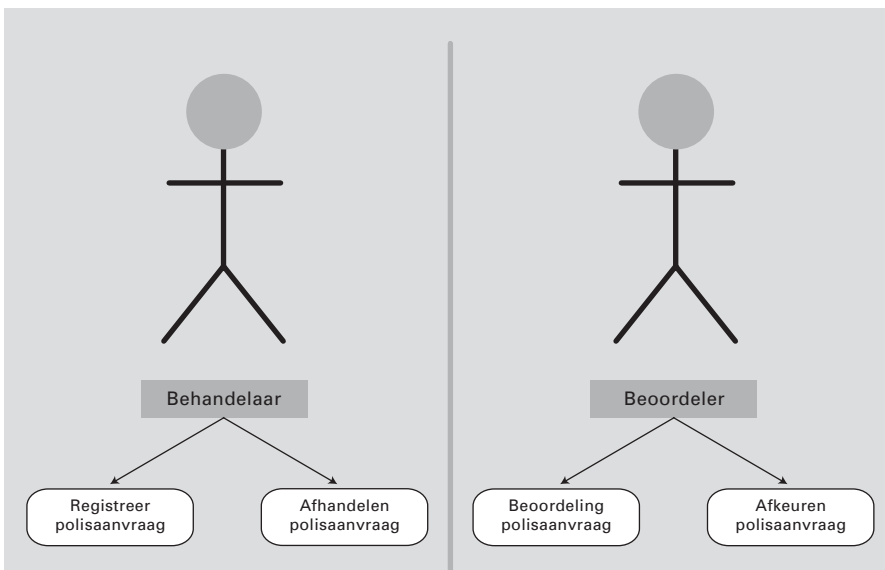
symbolen voor activiteiten, toestanden, toestandsovergangen en synchronisaties. In principe kom je een heel eind met deze symbolen. Uitbreiding van de symbolen is soms aan te bevelen. Dit is mogelijk door middel van het definiëren van stereotypen van bestaande symbolen. De toevoegingen die bruikbaar kunnen zijn zijn: termijnbewaking, inkomende en uitgaande triggers en wachtmomenten. Deze toevoegingen stellen de organisatie bijvoorbeeld in staat om door middel van een automatische vertaler het model te importeren in een simulatiepakket. De boodschap is hier: kijk welk doel er moet worden bereikt met de specificaties en welke standaarden en richtlijnen daarvoor nodig zijn.

#### *Traceability*

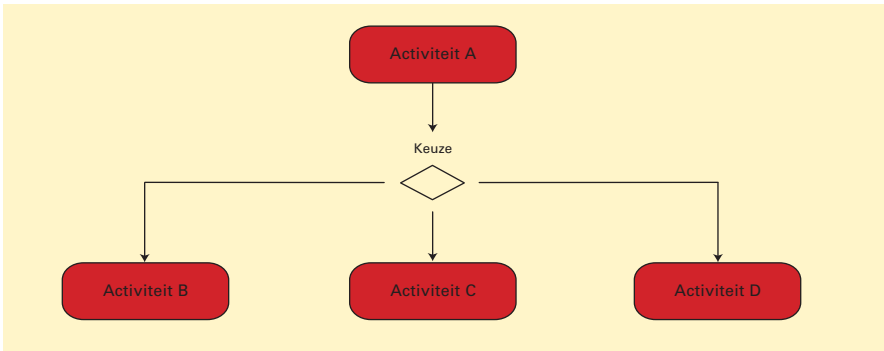
Door in Rational Rose activiteiten aan use-cases te koppelen wordt vanaf hier het ontwerp traceable. Het is namelijk niet mogelijk om met het tool activiteiten binnen een activity-diagram te linken aan andere concepten binnen het tool. In een use-case-diagram (zie afbeelding 2) worden per proces alle use-cases en actoren gemodelleerd. Vanaf hier is het ontwerp tot op serviceniveau traceable.

#### *Use-cases*

In afbeelding 3 is te zien dat na activiteit A een logistische keuze volgt voor activiteit B, C of D. Deze uitkomst moet terugkomen in de use-casebeschrijving van activiteit A. Preciezer gezegd, in de postconditie van de use-case moet de uitkomst zijn opgenomen op basis waarvan de keuze zijn 'beslissing' maakt. Use-cases bevatten in hun tekst ook regelmatig uitzonderings- of alternatieve scenario's. Zolang dit past binnen de afbakening van een activiteit is dit prima, anders moet men hier aparte use-cases voor schrijven die weer aan andere activiteiten in het activity-diagram zijn te relateren.



Afbeelding 2. Een use-case-diagram.



Afbeelding 3. Een keuzemoment.

### Ontwerp voor workflowimplementatie

Wanneer het ontwerp als basis dient voor een workflowimplementatie is het goed om in het componentontwerp en de sequence-diagrammen nog wat extra informatie toe te voegen. Wanneer een proces door een workflow-engine wordt ondersteund, wordt er een procesrecord bijgehouden door de workflow-engine. In dat procesrecord staat de informatie die nodig is om werk te kunnen routeren, te identificeren in werkbakken en om de koppeling te kunnen leggen tussen het proces en de materiegegevens in het informatiesysteem. In het ontwerp wordt voor elk gedefinieerd proces het procesrecord gemodelleerd in de vorm van een component met data-elementen en services. Voor elke activiteit wordt een sequence-diagram gemodelleerd. Dit sequence-diagram toont wanneer het procesrecord in de vorm van een component wordt geraadpleegd of gewijzigd naar aanleiding van communicatie tussen de workflow-engine en het informatiesysteem.

### Open houding

De werkwijze die in dit artikel is beschreven is geschikt voor het modelleren van systemen die procesgerichte ondersteuning bieden. Ad hoc functionaliteit of beheermogelijkheden van stamgegevens worden niet in processen gemodelleerd. Hiervoor zijn de standaard UML-technieken en afbakeningscriteria te gebruiken. Ervaring leert dat de discussie rondom het combineren van

technieken niet altijd even gemakkelijk gaat. Het vraagt om een open houding ten opzichte van methoden en technieken die binnen de verschillende ICT-disciplines worden gebruikt. Uitgangspunt moet niet de methode zijn, maar het doel dat je ermee wilt bereiken.



(Advertentie)

### Noot

1. UML (Unified Modelling Language) is een algemene visuele modelleertaal die is bedoeld om de verschillende producten in een systeemontwikkelproject te specificeren, te ontwikkelen en te documenteren.

### Literatuur

1. Fowler, Martin, UML Distilled: A brief guide to the standard Object Modeling Language, 2e editie, Addison-Wesley, ISBN: 0-201-65783-X.
2. Hoogendoorn, Sander Pragmatisch modelleren met UML 2.0, Addison Wesley, 2004, ISBN 90-430-0652-1.
3. Baardman, Eric, Oog voor de organisatie, Business Process Magazine 6, jaargang 7, oktober 2001, pagina 34-41.
4. Joosten, Stef (en anderen), Praktijkboek voor Procesarchitecten, Kon. van Gorcum, Assen, Sept. 2002.

### **Eric Baardman en Bas Huis in 't Veld**

*Eric Baardman is consultant bij Ordina Enterprise Solutions en Bas Huis in 't veld is consultant bij Ordina Finance.*