

Als traditionele processen niet meer voldoen

# FLEXIBELE PROCESBESCHRIJVINGEN

Als je boeken of artikelen over het modelleren van bedrijfsprocessen open slaat, zie je netjes uitgewerkte diagrammen van processen, met condities, alternatieven en flowlanes. Voor gevorderden komen er nog messages en compensation bij. Deze methode werkt heel vaak goed. Maar wat als je er tijdens het modelleren achter komt dat je model veel meer details bevat dan je in eerste instantie dacht? Of als je merkt dat je model uit wel heel erg abstracte processtappen bestaat?

Door Dick van Soest

Dit artikel gaat in op de vraag welke soorten flexibiliteit je kunt onderkennen, hoe je dit kunt herkennen, en ik beschrijf kort een aanpak om bedrijfsprocessen en werkprocessen flexibel te modelleren. Belangrijk gegeven is bovendien dat als je processen flexibel wil modelleren, de kennis in het systeem ook flexibel beschreven moet worden: declaratief in plaats van procedureel (zie kader). Kennis kan op meerdere manieren worden vastgelegd, zoals in regels, tabellen, beslisbomen en cases. Het hangt van de toepassing af welke representatievorm het beste is.

## Niveaus van flexibiliteit

Om te begrijpen waarom een andere aanpak nodig is dan de standaard manier van tekenen van processen, deel ik processen in in vier niveaus. Op het eerste niveau kunnen processen heel goed beschreven worden op de gebruikelijke manier, maar als de processen complexer worden volstaat dat niet meer en is een andere beschrijvingswijze beter op zijn plaats, zie afbeelding 1. De processen moeten dan op een meer flexibele manier beschreven worden. De processtappen zijn niet 'vast' aan elkaar gekoppeld, maar op executietijd wordt

bepaald welke volgende stap gezet gaat worden. Naarmate de procesbeschrijving flexibeler wordt, en het systeem meer kennis bevat, moet die kennis op een beter te onderhouden manier worden vastgelegd.

Het begrip systeem dat in dit artikel gebruikt wordt kan op een computersysteem slaan, maar kan ook een proces zijn dat in een organisatievorm is vastgelegd.

## Niveau 1: rigide processen

Het meest herkenbaar zijn de volgens een vast stramien verlopende processen. Voorbeelden te over: verkoopadministratie, afhandeling van hypotheekaanvragen, reserveringsprocessen. Kenmerkend is dat het proces een beperkt aantal vaste paden zal doorlopen. Dit is lang niet altijd eenvoudig te modelleren. In de context van BPM zal de organisatie waarin het proces gebruikt wordt voortdurend bezig zijn zich aan te passen aan de veranderende eisen van de markt of de doelstellingen van het bedrijf. In ieder geval is een kenmerk van dit soort processen dat op basis van gevalskenmerken het altijd mogelijk is een keuze te maken welk pad gevolgd gaat worden. Het proces is van te voren volledig uit te tekenen. Wordt het moeilijker, dan kun je altijd nog een processtap inbouwen waarin een ervaren medewerker handmatig wat gegevens aanpast. Op dit

niveau moet het systeem kennis hebben over welke stappen toegestaan zijn.

Er is op dit niveau niet bijzonder veel kennis in het systeem aanwezig. Het systeem heeft kennis nodig om te bepalen welke vervolgstappen moeten worden gezet. Omdat dit type proces van te voren goed te modelleren is, is het mogelijk om te sturen op efficiëntie en toch (of misschien juist) hoge kwaliteit te bieden aan degene voor wie het proces uiteindelijk uitgevoerd wordt: de klant.

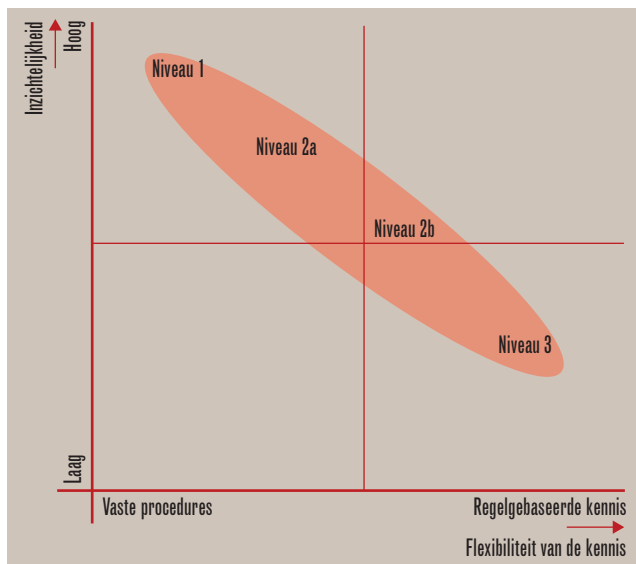
### Niveau 2a: groot aantal vaste vervolgpaden

Extra flexibiliteit kan geboden worden door de gebruiker van het systeem mee te laten bepalen hoe het proces verloopt. Dat is nodig op het moment dat het lastig of misschien zelfs onmogelijk is om op basis van het onderhavige geval te bepalen wat de vervolgstap moet zijn. Een voorbeeld is het proces van het afhandelen van claims van verkeersschade. Hierbij kunnen veel partijen betrokken zijn: andere verkeersdeelnemers, politie, Rijkswaterstaat, gemeente. Afhankelijk van wat er precies gebeurd is, en welke partijen erbij betrokken zijn, verloopt het proces anders. Een ander voorbeeld is het afhandelen van schade- of onderhoudsmeldingen bij een woningbouwmaatschappij. Dit is een proces dat afhankelijk van de vraag van de eindgebruiker steeds op een andere manier afgehandeld moet worden: de gebouwbeheerder kan ingeschakeld worden, een klusbedrijf of een aannemer. Er zijn vaste oplos-

## Procedurele versus declaratieve kennis

Er zijn twee manieren om kennis vast te leggen: procedureel of declaratief. Wanneer kennis procedureel wordt vastgelegd, betekent dit dat er een 'recept' wordt gegeven om een gegeven te berekenen. Dit recept hoeft alleen maar gevolgd te worden, stap voor stap. Een typisch voorbeeld van procedureel vastgelegde kennis is een subroutine in een taal als Java of C#.

Declaratief vastgelegde kennis beschrijft de verbanden tussen gegevens, zonder een rekenrecept te geven. Een voorbeeld is 'een persoon met een leeftijd ouder dan 18 jaar is volwassen'. Dit beschrijft het verband tussen de leeftijd van een persoon en het begrip volwassen. Een ander stuk declaratieve kennis is 'de leeftijd van een persoon is het aantal gehele jaren tussen de huidige datum en de geboortedatum'. Deze twee stukken kennis kunnen gecombineerd worden om de volwassenheid van een persoon te bepalen op basis van zijn of haar geboortedatum. Dit kan ook andersom gebruikt worden: uit de volwassenheid van iemand kan worden afgeleid dat hij of zij minstens 18 jaar geleden geboren is.



Afbeelding 1: Flexibiliteit versus inzichtelijkheid.

singen, maar het is niet duidelijk welke oplossing wanneer ingezet moet worden, dat hangt van teveel variabelen af. Het is lastig de hier geschetste processen op een vaste manier te modelleren. Het is niet onmogelijk, maar het is niet handig omdat er veel mogelijke paden zijn. Een uitweg daaruit binnen de gebruikelijke modelleringparadigma's is om de processtappen te abstraheren: processtappen die in veel takken van het proces voorkomen worden samengenomen. Het gevolg is minder ondersteuning voor de gebruiker.

Een geheel andere manier van modelleren, die meer recht doet aan de aard van dit soort problemen is om in het deel van het proces dat teveel takken bevat, de gebruiker keuzes te laten maken voor de vervolgstappen. In deze variant ondersteunt het systeem de gebruiker hierbij niet, maar kiest hij of zij zelf welk pad of welke paden voor het onderhavige geval van toepassing zijn. Dit kan zo simpel zijn dat het systeem een aantal mogelijke vervolgstappen voorstelt, en de gebruiker daaruit kiest. Een keuze voor een procespad houdt in dat de processtappen die daar onderdeel van uitmaken, aan de werklijst worden toegevoegd. Er moet wel rekening worden gehouden met het feit dat mensen maar een beperkte hoeveelheid keuzevariabelen tegelijk kunnen gebruiken. Het systeem heeft op dit niveau van modelleren beperkte kennis nodig: naast de vaste overgangen die op niveau 1 al bestonden heeft het systeem kennis over welke mogelijke vervolgpaden er zijn, maar niet zoveel kennis dat het systeem zelf het vervolg kan bepalen.

### Niveau 2b: groot aantal vaste vervolgpaden

Als de gebruiker niet in staat is om een goede afweging te maken kan ervoor gekozen worden om het systeem de gebruiker te laten ondersteunen in het kiezen van het juiste pad om te doorlopen. Het kan zijn dat de gebruiker onervaren is en

daarom geen keuzes kan maken, of dat de afwegingen te complex zijn. Voor de keuze van de vervolgstappen kunnen bijvoorbeeld SLA's van belang zijn, maar ook kosten voor de organisatie, of de kwaliteit die geboden wordt aan de klant. De overwegingen die hierin meegenomen worden zijn secundaire doelen van de organisatie, die onderling conflicterend kunnen zijn.

Op zijn minst kan het systeem de gebruiker helpen om een keuze te maken door een ordening aan te geven in de verschillende paden, in het uiterste geval kiest het systeem de beste oplossingsrichting. Op dit niveau van complexiteit is van te voren bekend welke vervolgpaden er zijn en kan het systeem de gebruiker volledig ondersteunen in de keuze voor het vervolgpad.

Het systeem heeft meer kennis over het domein dan in niveau 2a. Wat betreft de modellering, het opzetten van het systeem, vereist dit niveau daarom ook meer inspanning. De afwegingen die gemaakt moeten worden door het systeem, moeten begrepen kunnen worden om gemodelleerd te worden. De afweging ligt in de kosten van het modelleren (uitgedrukt in tijd of geld) ten opzichte van de kosten die worden gemaakt tijdens het gebruik van de processen. Op dit niveau is het al bijna niet meer mogelijk om deze kennis procedureel vast te leggen. Declaratieve kennis begint hier voordeel op te leveren: je legt kennis over het domein vast, het systeem zoekt wel uit hoe de kennis toegepast moet worden.

### Niveau 3: geen vaste vervolgpaden mogelijk

Hier zijn we op het hoogste niveau van flexibiliteit aanbeland. Op dit niveau is het niet meer mogelijk (of beter gezegd: efficiënt) om een vast aantal paden door het proces vast te leggen. Het is ook niet meer goed mogelijk (of efficiënt) om voor een deel van het proces een deelpad vast te leggen. Het onderhavige geval is zo complex dat de gebruiker niet kan overzien hoe het einddoel moet worden bereikt.

Voorbeelden van situaties waarin een extreem flexibele procesbeschrijving nodig is, zijn adviestrajecten. Dit zie je bijvoorbeeld bij het samenstellen van een behandelplan voor een patiënt met een complex ziektebeeld. Een ander voorbeeld is het afhandelen van meerdere, niet te isoleren verstoringen tegelijk in het spoorwegnetwerk. In dit soort situaties is een keuze voor één standaardoplossing niet meer mogelijk, zeker omdat de standaardoplossingen elkaar kunnen beïnvloeden. Als de workflows alleen maar voorzien in standaard oplossin-

gen, zou de gebruiker in een dergelijke situatie terug moeten vallen op zijn eigen kennis om een goed plan te maken om de situatie op te lossen.

Een methode voor het omgaan met dit soort processen is het systeem een goed pad te laten samenstellen, in plaats van een vervolgpad te kiezen. Daarmee is het mogelijk om een relatief onervaren gebruiker in een complex domein te laten werken of om een ervaren gebruiker het systeem te laten gebruiken met minder tijdsinspanning en minder kans op fouten. Een oplossing voor modellering op dit niveau van complexiteit is om de stappen in een proces als losse elementen te beschrijven. Bij ieder onderdeel wordt beschreven welk 'doel' de processtap heeft. Het systeem stelt voor ieder geval dynamisch het beste vervolgpad samen.

Bedrijfsprocessen worden meestal op meerdere lagen beschreven, variërend van het bedrijfsproces tot het werkproces. Deze flexibele manier van modellering kan op alle lagen gebruikt worden die onderkend worden in de procesbeschrijving. Op iedere laag van het bedrijfsproces worden de doelen met de juiste abstractie beschreven. Het is echter de vraag of flexibiliteit op alle lagen nodig is. Het zou kunnen dat de bedrijfsproceslaag bijvoorbeeld nog statisch beschreven kan worden. Op lagen met meer details zouden de procesonderdelen los gemodelleerd kunnen worden en het procespad wordt dan dynamisch samengesteld.

### Procespad

Deze manier van modelleren levert een grote oplossingskracht, maar vraagt ook veel inspanning om correct te modelleren. Het systeem moet nu voldoende kennis hebben om een procespad samen te stellen en te kunnen redeneren over de doelen die met een proces worden bereikt. De algoritmes hiervoor zijn gelukkig al wel bekend: in de kennistechnologie zijn zoekalgoritmes standaard. De kennis die hiervoor nodig is kan op dit niveau echter niet procedureel worden vastgelegd, zelfs objectgeoriënteerde talen zijn hiervoor niet flexibel genoeg. De kennis dient te worden vastgelegd in de vorm van kennisregels of beslistabellen om voldoende uitdrukingskracht te hebben.

Voor het opstellen van een behandelplan voor een patiënt met meerdere ziektes kan niet worden volstaan met simpelweg de behandelplannen voor de afzonderlijke ziektes te combineren. Er kan interactie ontstaan tussen de verschillende behande-

Niveau	Proces	Kennis	Geschiktheid voor soort proces
1	Vaste processen	Weinig kennis	Administratief, relatief eenvoudig
2a	Groot aantal vaste oplossingspaden	Weinig kennis	Complex, maar wel overzichtelijk
2b	Groot aantal vaste oplossingspaden	Meer kennis	Complex, voor mensen niet meer direct inzichtelijk
3	Alleen processtappen zijn vooraf bekend	Veel kennis	Zeer complexe problemen

Afbeelding 2: Overzicht van de vier lagen van flexibiliteit.

lingen, zoals medicijnen die niet met elkaar gecombineerd mogen worden. Er moet ook rekening gehouden worden met de lichamelijke en psychische belasting van de patiënt. Aan de andere kant zou het ook kunnen dat één behandeling een positief effect heeft op meerdere ziektes.

Door alle factoren uit te modelleren wordt de kennis die meestal impliciet in een behandelplan zit, expliciet gemaakt. Een process engine kan deze kennis gebruiken om een behandelplan samen te stellen, en daarbij beter rekening houden met de combinaties van factoren die voor een mens niet meer te overzien zou zijn. Op dit niveau is procedurele kennis niet meer bruikbaar. Het gaat hier juist om het combineren van kennis over verschillende aspecten, waar declaratieve kennis zeer geschikt voor is.

### De vier niveaus

Naarmate we omhoog gaan in de niveaus neemt de flexibiliteit van de processen toe. De benodigde kennis in het systeem neemt toe, en daarmee ook de modelleringsinspanning. Het systeem gaat op de hogere niveaus meer ondersteunen tijdens de uitvoering bij het vaststellen van de volgende processtap. We noemen dit 'guided activities'. Voor de gebruiker van het systeem betekent dit een betere ondersteuning, maar daarmee kan de inzichtelijkheid tijdens de modelleringsfase juist afnemen als niet de goede tooling wordt gekozen. De kennis moet op het juiste abstractieniveau beschreven worden, bijvoorbeeld in de vorm van executeerbare business rules of beslistabellen.

Vanwege het spanningsveld tussen flexibiliteit en inzichtelijkheid moet er een gedegen afweging gemaakt worden of dergelijke flexibiliteit ook echt gewenst is. Een flexibele oplossingsrichting moet alleen gekozen worden als er een noodzaak voor is, maar dan biedt deze oplossing ook duidelijke meerwaarde. Een overzicht van de vier niveaus staat in de tabel in afbeelding 2.

### De voordelen

Er zijn vier niveaus van complexiteit beschreven waarop processen kunnen worden gemodelleerd. Naarmate de complexiteit van het probleem groter wordt, is er krachtiger ondersteuning van het systeem nodig om een gebruiker door het proces te leiden. Dat vereist dat er meer kennis in het systeem zit, wat een grotere inspanning vraagt tijdens het bouwen van het systeem. Wat er mee gewonnen wordt, is dat het systeem op het juiste niveau van abstractie gemodelleerd wordt. Dit betaalt zich terug in lagere kosten van onderhoud. Wat je er ook mee wint is betere ondersteuning van de gebruiker. Hierdoor kan een minder ervaren gebruiker toch complexe gevallen afhandelen of kan een ervaren gebruiker meer werk verzetten met minder fouten.

Naarmate de niveaus oplopen verandert ook de inzichtelijkheid. Op de eerste twee niveaus is er nog sprake van vaste paden, ook al is het niet van te voren bepaald welk pad op welk moment gekozen gaat worden. Op niveau 3 zijn er zelfs

## Declaratieve kennis in procesbeschrijvingen

Declaratief beschreven kennis is beter te onderhouden dan procedurele kennis. Bij de declaratieve beschrijving worden relaties vastgelegd tussen gegevens. In een goed opgezet systeem hebben deze gegevens betekenis voor de mensen die de kennis onderhouden. In een systeem met declaratieve kennis is het geen probleem om 'kleine' stukken kennis vast te leggen, zoals de relatie tussen de geboortedatum en de leeftijd. Het inferentiemechanisme zorgt ervoor dat de juiste kennis aan elkaar geknoopt wordt om een benodigd gegeven uit te rekenen. Het 'uit elkaar trekken' van de kennis tot kleine brokjes leidt ertoe dat het beter te onderhouden is. Ieder stuk kennis kan op zichzelf gezien worden, heeft een duidelijke betekenis, en kan apart onderhouden worden.

Een ander voordeel van declaratieve beschrijvingen is dat ze begrijpelijker zijn dan procedurele beschrijvingen. Vergelijk een procedurele routebeschrijving in termen van "Honderd meter noordelijk, rechtsaf slaan, twee kilometer rechtdoor, linksaf, ..." met een declaratieve beschrijving als: "Rijd naar het noorden tot het Paulusplein, vandaar de Rechterstraat in tot de kruising met de Boslaan, ...". De procedurele beschrijving voldoet zolang die precies gevolgd kan worden. Als er iets mis gaat onderweg is de procedurele beschrijving nutteloos geworden. Het is voor mensen gemakkelijker om een relatie te begrijpen, dan om een procedure af te werken en daaruit af te leiden wat het effect is op de uitvoergegevens.

geen paden meer, maar alleen activiteiten die aan elkaar geknoopt worden als er een vervolgpad bepaald moet worden. Zeker op niveau 3 is het daarom lastig om tijdens het modelleren inzicht te krijgen in de uiteindelijke werking van de processen. Daarom is het noodzakelijk om – bij het modelleren van processen op deze manier – tools te gebruiken die inzicht kunnen geven in de mogelijke paden die kunnen worden afgeleid. Dit kan door testcases te gebruiken, maar de tools moeten ook inzicht geven in hoe de doelen samenhangen. Ook is het voor systemen die veel sturing geven aan gebruikers belangrijk dat het gekozen pad aan de gebruiker uitgelegd kan worden. Het blijkt dat gebruikers het moeilijk vinden om oplossingen te accepteren die door een systeem worden voorgesteld. Dit geldt zeker als het systeem voor hen nog onbekend is. Net als personen moet een systeem zich eerst bewijzen voordat de conclusies geaccepteerd worden.

Dick van Soest

Dr. Ir. D.C. van Soest is business engineer bij Everest.