

Er wordt tegenwoordig veelvuldig gesproken over bedrijfsprocessen en de besturing daarvan door software. Het gebruik van veel verschillende begrippen zorgt voor verwarring, maar is bij nader inzien niet zo vreemd omdat besturing in verschillende softwarelagen voorkomt. Dit artikel gaat verder in op de rol van software bij het besturen van processen.

achtergrond

Een model voor procesondersteuning in software

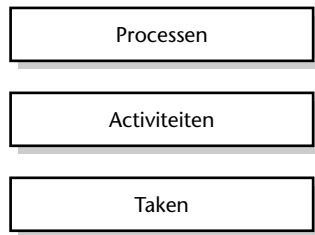
Scheiden van functie- en sturingsaspecten

Na een periode van stilte is er hernieuwde aandacht voor de automatisering van bedrijfsprocessen in software. Zo bieden allerlei standaardpakketten voor bijvoorbeeld ERP en CRM workflow mogelijkheden. Daarnaast is er een sterke beweging om processen met middleware te ondersteunen, waarbij wordt gesproken over EAI, “message brokers” en “integration brokers”. Dit gaat samen met de aandacht voor webservices, die bedrijven in staat stellen processen beschikbaar te stellen aan andere bedrijven. Naast het gevestigde begrip “workflow management” wordt nu ook gesproken over “business proces management”, “case management” en “orchestration”. In dit artikel zal duidelijk worden dat deze begrippen te maken hebben met softwarematige besturing van processen. Na een korte inleiding in de wereld van processen wordt een model gepresenteerd waarin verschillende vormen van sturing door software worden beschreven.

VIER-OGEN PRINCIPE Wat zijn nu eigenlijk bedrijfsprocessen? Vanuit een IT perspectief zien we een transformatieproces van invoergegevens naar uitvoergegevens, waarbij tevens uitvoerenden (mensen of systemen) worden aangestuurd. Een bedrijfsproces kent dus twee aspecten: het functionele en het sturingsaspect. Het functionele aspect beschrijft de transformatie van invoer naar uitvoer, bijvoorbeeld door het uitvoeren van een berekening. Het sturingsaspect beschrijft op welk moment in tijd, in welke volgorde en door wie processen worden uitgevoerd. Denk bijvoorbeeld aan het afdwingen van een vier-ogen principe. Processen kunnen door mensen worden uitgevoerd, maar ook

gedeeltelijk of volledig door geautomatiseerde systemen. Processen kunnen verschillend van aard zijn. Enerzijds langdurig en complex, waarbij verschillende mensen en systemen een bijdrage moeten leveren. Anderzijds kunnen zij ook kort van duur zijn en door één mens of systeem worden uitgevoerd. Lange tijd was de besturing het onderwerp van alleen managers en bedrijfskundigen. Tegenwoordig is de IT-inrichting bepalend: de snelheid, transparantie en automatiseringsgraad van efficiënt uitgevoerde bedrijfsprocessen bepalen de concurrentiepositie.

KREDIETAANVRAAGPROCES Het is handig voor verschillende soorten processen ook verschillende termen te gebruiken. In figuur 1 wordt een drielagen model voor processen beschreven, waarbij onderscheid wordt gemaakt tussen processen, activiteiten en taken. Processen zijn langlopend en worden veelal door meerdere mensen en systemen uitgevoerd. Ze plaatsen uit te voeren activiteiten in een logische volgorde en specificeren uitvoerende rollen en verantwoordelijkheden. Een activiteit is een eenheid van tijd, plaats en handeling en kan ononderbroken door één persoon of systeem worden uitgevoerd. Aan het einde van een activiteit wordt werk veelal overgedragen aan een ander persoon of systeem. Binnen een activiteit worden één of meer taken uitgevoerd, waarbij de volgorde van taken vaak niet van belang is. Een taak is een hoeveelheid werk die binnen een korte hoeveelheid tijd wordt uitgevoerd en die altijd in zijn geheel moet slagen of falen. Een voorbeeld van een proces is het kredietaanvraagproces waarbij naast de opsteller van de kredietaanvraag



FIGUUR 1. Proceslagen

ook een fiatur betrokken is. Eén van de taken bij het opstellen van de kredietaanvraag is het uitvoeren van een kredietwaardigheidstoets bij het BKR.

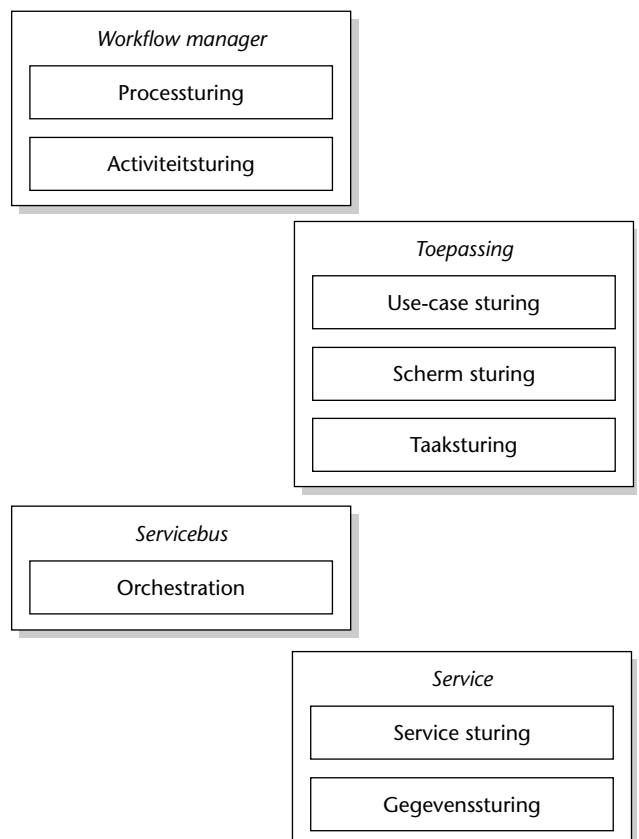
Wat is nu de rol van software bij het uitvoeren van processen? Zoals gezegd heeft een proces een functie- en een sturingsaspect. Het is dan ook logisch dat de rol van software tweeledig is. In de eerste rol is software verantwoordelijk voor het uitvoeren van transformaties (functie). In de tweede rol bestuurt software het proces. De nadruk in dit artikel ligt op de sturingsrol van software bij het uitvoeren van processen. Deze rol is afhankelijk van de uitvoerder van het proces. Besturing van menselijke uitvoerders ondersteunt het uitvoeren van processen door te bepalen welke activiteiten en taken er op een bepaald moment kunnen worden uitgevoerd. Besturing van processen waarvan de activiteiten door software worden uitgevoerd geschiedt volledig automatisch. Deze twee vormen van sturing delen de eigenschap dat bij het optreden van een gebeurtenis moet worden bepaald welke volgende activiteit of taak moet worden uitgevoerd en door wie (of wat). Deze keuzen kunnen van tevoren zijn vastgelegd of kunnen zijn gebaseerd op de huidige toestand van het proces. In dat laatste geval zijn er regels nodig die werken op de procestoestand, zoals condities of algoritmen.

Het streven is om de functie- en sturingsaspecten in afzonderlijke softwarecomponenten te definiëren. Hierdoor is het hergebruik van functionele componenten in een andere context eenvoudiger. Doordat sturing apart is gedefinieerd kan het ook eenvoudiger worden gelokaliseerd en aangepast waardoor de onderhoudbaarheid en herbruikbaarheid van software toeneemt. Het blijkt dat er verschillende vormen van sturing bestaan, mede doordat er verschillende dingen bestuurd moeten worden. Om te bepalen welke vormen van sturing dat zijn is het dus noodzakelijk eerst de functionele componenten te kennen. Het probleem is dat hier geen universeel antwoord voor bestaat: het is afhankelijk van de software architectuur, waarin de verschillende soorten componenten zijn gedefinieerd.

In dit artikel wordt een generieke software architectuur gehanteerd waarin verschillende vormen van stu-

ring worden gepositioneerd. Deze software architectuur bestaat uit componenten in een gelaagde structuur. In elk component in de software architectuur bestaan een aantal verschillende vormen van sturing, die zelf ook een lagenstructuur vormen. Deze sturingsvormen zijn vaak ook expliciet als component herkenbaar. De lagenstructuur van componenten en sturingsvormen geeft weer hoe aanroepen typisch plaats vinden; van boven naar beneden. Ook neemt de duur van de processen die in de sturingslagen worden ondersteund af van boven naar beneden. De lagenstructuur moet niet te strikt worden genomen; lagen kunnen worden overgeslagen en een aanroep in een laag zou ook kunnen resulteren in een aanroep naar een hogere laag. Hierdoor is het bijvoorbeeld mogelijk dat een kortlopend proces een langlopend proces opstart. Sommige componenten in de architectuur hebben niet veel meer dan een sturende rol, terwijl in andere componenten wel degelijk ook functionele componenten te herkennen zijn.

Het resulterende sturingsmodel is weergegeven in figuur 2. Componenten in dit model zijn workflow manager, toepassing, servicebus en service. Workflow managers zijn gericht op het sturen van langlopende processen en activiteiten. Toepassingen zijn functionele componenten waarmee gebruikers activiteiten kunnen uitvoeren. Zij maken gebruik van herbruikbare services waarin bedrijfslogica en -gegevens zijn gedefinieerd.



FIGUUR 2. Softwaresturingsmodel

Een servicebus ontkoppelt de toepassingen van de services. In de volgende paragrafen worden de componenten in het model verder beschreven.

Workflow managers en modernere case-managers besturen processen en activiteiten. Deze sturing wordt vastgelegd in de vorm van procesdefinities waarin staat wat de volgorde, routering, autorisatie en inhoud van de uit te voeren activiteiten is. Bij veel workflow management tools ligt de nadruk op ondersteuning van processen die door gebruikers worden uitgevoerd, maar er zijn ook tools die langlopende geautomatiseerde processen ondersteunen. Processen van de eerste soort worden op verzoek van gebruikers gestart, waarna vervolgvormen uit een werkbak te selecteren zijn. Het starten van een proces resulteert in een procesinstantie (ook wel "case" genoemd) waarin de procesgegevens worden geadministreerd. Softwarecomponenten die bepaalde taken in een activiteit uitvoeren worden automatisch gestart en geven hun resultaat terug aan het workflow management tool die dit verwerkt in de processtatus. Zo'n interactie met een softwarecomponent wordt ook wel een "use-case" genoemd¹; een interactie met een systeem dat een voor de afnemer zinvol resultaat oplevert. Bij activiteiten die volledig handmatig worden uitgevoerd moet de gebruiker het tool expliciet vertellen dat de activiteit is afgerond. Case-management gaat verder dan workflow management door ook inzicht te geven in de voortgang binnen een activiteit.

TOEPASSINGEN Toepassingen ondersteunen gebruikers in het uitvoeren van activiteiten. Zij bevatten de realisatie van één of meer use-cases, die worden geactiveerd door een workflow manager. Een use-case bestaat uit een opeenvolging van schermen en taken die worden geactiveerd op basis van handelingen van de gebruiker. Veelal wordt de sturing van een individuele use-case in een speciaal daarvoor ontwikkeld component in een systeem ondergebracht. Zo'n component wordt een "use-case controller" genoemd. In het geval van een Java-gebaseerde web applicatie zal zo'n controller typisch gerealiseerd worden als servlet. Binnen een use-case kunnen verschillende schermen worden gebruikt. Het besturen van schermen is vaak expliciet belegd in een softwarecomponent binnen het systeem; een zogenaamde "interface controller". Deze weet welke schermelementen er bij een scherm horen en handelt alle gebruikershandelingen af die bij binnen de context van een scherm kunnen worden afgehandeld. Alle andere gebruikershandelingen worden teruggegeven aan de bovenliggende use-case sturingslaag. In web applicaties is vaak geen sprake van een expliciete interface controller en worden de verantwoordelijkheden

grotendeels ondergebracht bij de use-case controller. Handelingen van gebruikers hebben mogelijk het aanroepen van services tot gevolg, bijvoorbeeld voor het ophalen en verwerken van gegevens van schermen. De aansturing van die services die uit zo'n handeling voortvloeien noemen we taaksturing. Taken dienen meestal transactioneel te worden uitgevoerd; het half uitvoeren van een taak is onwenselijk. Taaksturing wordt typisch in een afzonderlijk softwarecomponent gerealiseerd. Een goed voorbeeld van zulke componenten zijn de "action" componenten die in het Struts raamwerk² moeten worden gedefinieerd. Struts is een raamwerk voor het ontwikkelen van webapplicaties conform het Model View Controller (MVC) model. Het kredietaanvraagproces wordt typisch ondersteund door een toe-

In een Java-gebaseerde webapplicatie zal de controller typisch gerealiseerd worden als servlet

passing, waarin meerdere use-cases beschikbaar worden gesteld. Eén van die use-cases biedt ondersteuning bij het opstellen van een kredietaanvraag. Binnen die use-case is er typisch sprake van meerdere schermen, bijvoorbeeld voor klantgegevens, productgegevens en jaarcijfers. Voorbeelden van taken die binnen deze schermen een rol spelen zijn het ophalen van de NAW-gegevens van een klant, het uitvoeren van een kredietwaardigheidstoets en het verzenden van de kredietaanvraag. Samenvattend bevatten toepassingen typisch drie vormen van sturing: use-case sturing, schermsturing en taaksturing.

Een service is een herbruikbare dienst met een gestandaardiseerde interface die wordt aangeboden door een softwarecomponent. Een service is bijvoorbeeld het opvragen van basisgegevens van een bepaalde klant. Het softwarecomponent dat de service realiseert is afhankelijk van andere componenten waarin de daadwerkelijke bedrijfslogica is gedefinieerd. Afhankelijk van de ontwikkelmethode kunnen dit procedurele componenten zijn of componenten die een bepaald bedrijfsobject representeren. Denk hierbij aan bedrijfsobjecten als "klant", "overeenkomst" en "factuur". Onafhankelijk van de aanpak zal er voor het realiseren van een service een verzameling operaties op deze componenten moeten worden aangeroepen en zal het resul-

1 In geval van een use-case die wordt gestart door een gebruiker wordt ook wel gesproken over een "business use-case"

2 Struts: <http://jakarta.apache.org/struts>

Nieuwe OptimalJ-versie gebruikt de kracht van patterns

Compuware heeft het product OptimalJ in de nieuwe versie 2.1 op de markt gebracht. OptimalJ is een Java-ontwikkelingsomgeving die in de voorhoede staat van de patroonbewerking. Patterns vergemakkelijken het hergebruik van vooraf gedefinieerde ontwerpen, structuren en code. Ze maken de specifieke kennis van enkele deskundige Java-architecten beschikbaar voor heel de organisatie. De functies voor het bewerken van patterns van OptimalJ geven softwarearchitecten de mogelijkheid om hun eigen patterns te ontwikkelen en te bewaren, om geautomatiseerd operationele J2EE-applicaties te creëren. Patterns leiden tot een snelle, gestandaardiseerde ontwikkeling in heel het ontwikkelingsproces - waarbij de ontwikkelaars zich op

het resultaat kunnen concentreren in plaats van op de methode. OptimalJ 2.1 geeft architecten een mechanisme om J2EE-patterns te definiëren die ontwikkelaars automatisch gebruiken wanneer ze applicaties construeren. Ze hoeven niet langer tijd en moeite te investeren in complexe, technische infrastructuren.

Modelgebaseerd ontwikkelen is het fundament van OptimalJ. Object Management Group's (OMG's) Model Driven Architecture (MDA) is de standaard die voor deze nieuwe ontwikkelmethoden gebruikt wordt. MDA verhoudt zich tot Unified Modeling Language (UML) op dezelfde wijze zoals J2EE zich verhoudt tot Java, of van .Net tot C#. In deze gevallen wordt een ontwikkeltaal gebruikt om een bepaalde architectuur te implementeren. OptimalJ is een van de eerste producten die gebaseerd is op MDA zoals deze is gedefi-

nieerd door de Object Management Group. OptimalJ breidt de MDA architectuur uit door het toevoegen van patterns aan de modellen. De functie voor patternbewerking vult OptimalJ aan met door de gebruiker gedefinieerde implementatiepatterns. Implementatiepatterns vertalen de applicatiemodellen van OptimalJ in fysieke code. Binnen OptimalJ worden de implementatiepatterns uitgedrukt in de Template Pattern-taal. Om patterns te kunnen ontwikkelen, omvat OptimalJ 2.1 een verzamelplaats zodat organisaties hun eigen patterns kunnen definiëren. Wanneer de architecten de J2EE-architectuur begrijpen, kunnen zij met behulp van OptimalJ zonder moeite hun eigen patterns opbouwen of bestaande patterns wijzigen.

Het gebruik van domeinpatterns, maakt het mogelijk om domeinmodellen voor nieuwe

toepassingen sneller te construeren en beperkt het aantal fouten in de modelleringsfase. Men kan ook een bibliotheek van domeinpatterns samenstellen en bijhouden. Daarnaast omvat OptimalJ 2.1 drie nieuwe diagrammen op het niveau van het applicatiemodel: het DBMS Relational Diagram, het EJB Component Diagram en het Web Component Diagram. Tenslotte zijn uitbreidingen voor presentatiemodellen inbegrepen, die ontwikkelaars de mogelijkheid geven om het formaat en de lay-out van attributen in het applicatiemodel te definiëren. OptimalJ 2.1 Professional Edition is onmiddellijk beschikbaar. Licenties op naam zijn verkrijgbaar vanaf 4.995 Amerikaanse dollar per ontwikkelaar. Er is ook een testversie beschikbaar die 14 dagen kan worden gebruikt. Meer informatie vind u op <http://www.compuware.nl>.

ADV. DELTA SOFTWARE

	proces sturing	activiteit sturing	use-case sturing	scherm sturing	taak sturing	orchestration	service sturing	gegevens sturing
looptijd	dagen	uren	minuten	minuten	seconden	seconden	milliseconden	milliseconden
gebruikers interactie	ja	ja	ja	ja	nee	nee	nee	Nee
communicatie	lokaal	lokaal of remote	lokaal	lokaal	remote	remote	lokaal	remote
technische transactie	nee	nee	nee	nee	optioneel	optioneel	ja	ja
typische vorm	workflow manager	case manager	use-case controller	interface controller	action controller	integration/message broker	service controller	persistentie tool

TABEL 1. Eigenschappen van sturingsvormen

taat moeten worden gecombineerd. Deze vorm van sturing noemen we servicesturing en zal typisch in een “service controller” component zijn gerealiseerd. Bij het uitvoeren van bedrijfslogica wordt gebruik gemaakt van bedrijfsgegevens die de toestand van de service beschrijven. Deze bedrijfsgegevens worden beheerd in een database; momenteel vaak nog een relationele database. Voor het inlezen of wegschrijven van gegevens zijn vaak meerdere verzoeken naar de database nodig. Dit is erg herkenbaar in objectgeoriënteerde benaderingen waarbij de toestand van een bepaald object verspreid kan zijn over meerdere relationele tabellen. Deze vorm van gegevenssturing staat dan ook bekend als objectrelationeel mapping, waarvoor speciale objectrelationele mapping tools beschikbaar zijn. Samengevat bevat een service minimaal servicesturing en gegevenssturing.

Een *servicebus* heeft als voornaamste doel toepassingen van services te ontkoppelen. Dat doet hij door logische verzoeken door te sturen naar de juiste service en aansluitend naar eventuele andere services. De verzoeken en antwoorden kan hij daarbij combineren en transformeren naar het juiste formaat. In feite maakt een dergelijke bus een nieuwe (complexe) service door bestaande services met elkaar te combineren. Een complexe service zou een integraal klantbeeld kunnen zijn die de basisgegevens van een klant combineert met bijvoorbeeld alle lopende overeenkomsten van de klant en de openstaande facturen. In de webservice architectuur zijn er reeds een aantal concurrerende standaarden die samenstellingen van services tot een nieuw proces kunnen beschrijven, zoals WSFL, XLANG en BPMI. Het moge duidelijk zijn dat het samenstellen van services vraagt om een speciaal soort sturing, wat tegenwoordig ook wel “orchestration” wordt genoemd. Dit is de markt van de integration brokers en message brokers.

Het is duidelijk dat er meerdere vormen van sturing bestaan, die elk zo hun specifieke eigenschappen hebben. Tabel 1 vat een aantal van die eigenschappen

samen, zoals de typische looptijd van de ondersteunde processen, aanwezigheid van gebruikersinteractie, lokaliteit van communicatie, ondersteuning voor (korte) technische transacties en typische vorm waarin de sturing zich manifesteert in software. Voor dat laatste aspect geldt dat veel sturing nu hard gecodeerd is in software, terwijl er standaard producten bestaan die hierin zijn gespecialiseerd. Gezien de specifieke eigenschappen van de sturingsvormen zullen dit vaak specifieke producten zijn. Onder de noemer “business process management tools” zijn er echter producten op komst die pretenderen goede ondersteuning te bieden voor meerdere sturingsvormen.

CONCLUSIES Bedrijfsprocessen hebben zowel functie- als sturingsaspecten en kunnen zowel door mensen als door software worden uitgevoerd. De onderkende vormen van sturing zijn afhankelijk van de software-architectuur. In dit artikel is aan de hand van een generieke software architectuur een model gepresenteerd voor verschillende vormen van sturing. Deze sturingsvormen hebben ieder hun specifieke eigenschappen, maar kunnen worden ondersteund door standaard producten. Kijk de volgende keer dus eens naar de toepasbaarheid van deze producten, maar zorg er minimaal voor dat sturingsaspecten in de software zijn gescheiden van functieaspecten.

Danny Greefhorst en Ferd Leliveld

Danny Greefhorst is werkzaam als IT Architect bij IBM Global Services en is bereikbaar onder greefhorst@nl.ibm.com. Ferd Leliveld is werkzaam als workflow coördinator bij de afdeling Bedrijf en Informatie Consultancy van ING en is bereikbaar onder ferd.leliveld@mail.ing.nl.