

Ontwikkelen met de Business Rules Engine van BizTalk Server 2004

EEN BUSINESS MODELLEER JE ALS BUSINESSPROCESSEN EN -REGELS

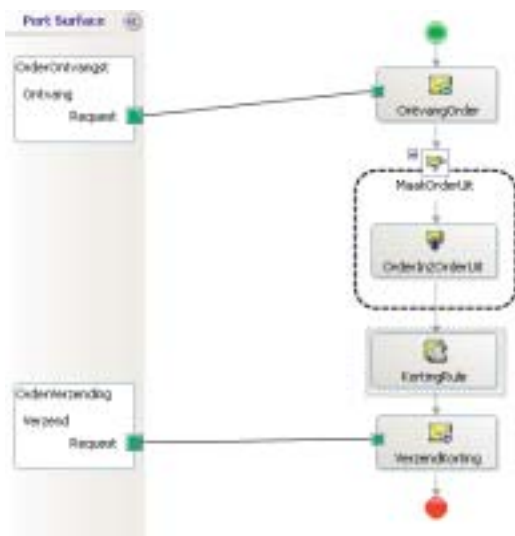
Een business modelleer je als businessprocessen en -regels. Businessprocessen beschrijven de high-level stappen die genomen moeten worden om een resultaat te verkrijgen voor een bepaalde klant of markt.

Businessprocessen zijn vrij statisch van aard. Businessregels bepalen en beperken de aspecten van de business die businessprocessen sturen. Zo kan een businessregel een businessproces sturen, bijvoorbeeld bij welke leverancier een bestelling wordt geplaatst. Businessregels zijn meestal complex van aard en worden geregeld aangepast om competitief te blijven met de concurrentie; kijk naar de financiële en verzekeringssector. Businessprocessen modelleer je in Microsoft BizTalk Server 2004 met orchestraties, en businessregels met de Microsoft Business Rules Engine. In dit artikel gaan we dieper in op de Business Rules Engine, en tonen we de synergie met BizTalk Server 2004 en .NET-applicaties.

Microsoft BizTalk Orchestrations

Microsoft BizTalk Server 2004 staat toe om businessprocessen te modelleren met behulp van orchestraties. Deze kunnen we bijvoorbeeld vergelijken met de UML-activiteitsdiagrammen, waar we de opeenvolgende acties modelleren die nodig zijn om een businessproces uit te voeren. Orchestraties laten ons toe om langlopende transacties te bouwen, waar we verschillende 'line-of-business'-applicaties kunnen oproepen met behulp van bijvoorbeeld web-services. Orchestraties zijn eerder statisch; ze wijzigen namelijk zelden omdat de businessprocessen zelf niet zo vaak veranderen. Een eenvoudig voorbeeld van een orchestratie om een korting toe te kennen aan een bestelling vind je in afbeelding 1. In deze orchestratie vind je ook een CallRules-shape die een businessregel aanroept. Deze regel bepaalt wanneer een korting wordt toegekend.

BizTalk 2004 orchestraties worden door ontwikkelaars aangepast, gecompileerd naar .NET-assemblies en vereisen dat ze telkens



Afbeelding 1. Orchestratie voor toekennen korting

worden geïnstalleerd. Deze stappen vergen veel tijd en kennis en zijn daarom enkel geschikt voor businessprocessen, omdat deze niet zozeer aan verandering onderhevig zijn. Businessregels die binnen dit framework worden aangemaakt, kunnen op eenvoudige wijze worden onderhouden en geïnstalleerd via de Business Rules Composer, zonder dat de bijhorende orchestraties hoeven worden aangepast of geïnstalleerd. Samen vormen BizTalk 2004 Orchestrations en de Business Rules Engine (BRE) een krachtige synergie om je business te modelleren. Nog een ander voordeel van de BRE is dat je deze ook kunt gebruiken in traditionele applicaties.

Business Rules Framework

Het Business Rules Framework bestaat uit de volgende onderdelen die door softwareontwikkelaars kunnen worden gebruikt:

- * Een krachtige rules-engine die wordt gebruikt om de verschillende businessregels te evalueren.
- * Omvangrijke APIs voor het ontwikkelen van applicaties in .NET die gebruik maken van de rules-engine.
- * De *Business Rule Composer*, een Windows-applicatie die door ontwikkelaars of businessanalisten gebruikt wordt om businessregels en vocabularia aan te maken of te onderhouden.
- * Een eenvoudige integratie van het Business Rule Framework in de orchestratie designer van BizTalk Server 2004 die je toelaat om vanuit een orchestratie één of meer businessregels te evalueren.
- * De *Rule Engine Deployment Wizard* die je toelaat om snel businessregels en vocabularia te importeren of te exporteren.

Vaak zul je ook de businessrules-engine willen monitoren en kijken welke businessregels er zoal afgevuurd worden en wat hun uitkomst is. BizTalk Server heeft ook *Health and Activity Tracking (HAT)*. Deze applicatie laat je toe de businessrules die werden uitgevoerd op te volgen en later te analyseren bij eventuele problemen.

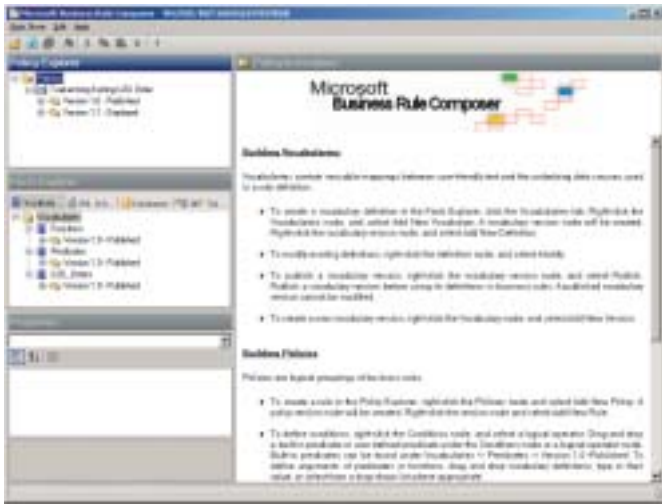
Businessrules-concepten

Laten we een eenvoudig voorbeeld van een businessrule gebruiken als voorbeeld en uitwerken in de Business Rules Engine. Hiervoor gaan we een businessregel gebruiken voor het toekennen van een korting aan een klant.

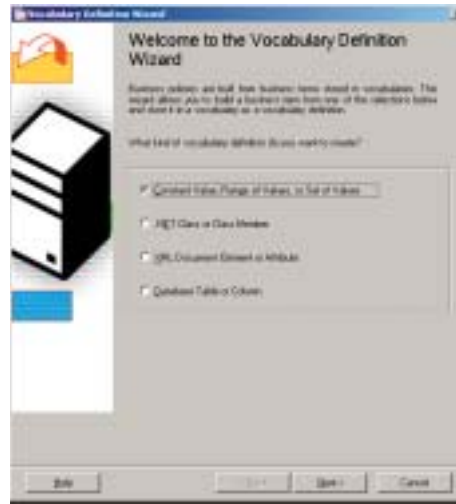
“Klanten die voor meer dan € 5000 aankopen doen krijgen een korting van 10%”

Zoals je ziet is een businessregel een conditie waaraan moet worden voldaan, gevolgd door een actie. Deze conditie kan één of meer feiten (totaalbedrag aankoop) controleren. Als aan deze conditie is voldaan, zal de actie uitgevoerd worden die dan bepaalde gegevens wijzigt: de korting.

IF conditie THEN actie



Afbeelding 2. De Business Rules Composer



Afbeelding 3. Aanmaken vocabularium

Dit soort businessregels kan je gebruiken om gegevens te wijzigen, maar ook om te beslissen of bijvoorbeeld een bepaalde order bevestigd moet worden, welke webservice aangeroepen zal worden, enzovoort. Een businesspolicy is een collectie van deze businessregels die als geheel worden toegepast. Om deze policies aan te maken, te wijzigen en te installeren gebruiken we de Business Rules Composer. We kunnen dit natuurlijk ook doen met behulp van .NET-classes uit het Business Rules Framework, maar dat valt buiten het onderwerp van dit artikel.

De Business Rules Composer

De Business Rules Composer is een applicatie waarmee we de businessregels aanmaken, wijzigen en installeren; zie afbeelding 2. Hierbij definiëren we de gegevens (facts) in de Facts Explorer pane die we gebruiken in onze businessregel. Tijdens de ontwikkeling van deze businessregels kunnen we ook testen of we wel het te verwachten resultaat verkrijgen.

Vocabularia

Om deze businessregel te implementeren in de Business Rule Engine moeten we eerst een mapping aanmaken tussen deze concepten en hun eigenlijke implementatie. Deze concepten worden beschreven in een zogenaamde vocabularium in het Facts Explorer deelvenster. Deze kan constanten, XML-documenten, databasegegevens en .NET-classes bevatten, die we in een vocabularium van een vriendelijke naam voorzien zoals beschreven in een businessre-



Afbeelding 4. Configuratie van de constante MinBedragKorting.

gel. De “€ 5000” bijvoorbeeld in het voorbeeld is een constante; het totaalbedrag van de aankoop is een XML-element en de korting is ook een XML-element. Als we een nieuwe definitie aanmaken - zie afbeelding 3 - krijgen we de keuze uit het aanmaken van een constante waarde, een element of attribuut uit een XML-document, een kolom uit een databasetabel of een .NET-classmethode.

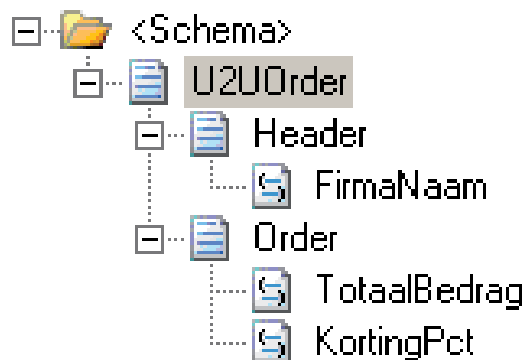
Voor het aanmaken van een constante kunnen we kiezen uit een constante, een interval of een verzameling van waarden. In ons voorbeeld is het MinKortingBedrag een constante met de waarde 5000. De configuratie zie je in afbeelding 4.

Voor het aanmaken van een fact die een XML-element beschrijft moeten we een XSD-schema kiezen, en daaruit het element of attribuut kiezen. Het XSD-schema dat we hier gebruiken is eenvoudig zoals afbeelding 5 aantoont. De configuratie van dit fact zie je in afbeelding 6.

Enmaal tevreden over het resultaat moeten we nog het vocabularium publiceren voordat we deze kunnen gebruiken.

Policies

Businessregels zelf worden in de Business Rules Engine aangemaakt in een policy. Policies groeperen verschillende regels en zijn de eenheid van deployment in de Business Rules Engine. Businessregels maken we aan door elementen van het vocabularium te slepen naar het regel-deelvenster. Hier bouw je expressies, samen met predikaten en functies. Verschillende predikaten kunnen met elkaar gecombineerd worden via de logische operator AND, OR en NOT. Zo kunnen we complexe expressies samenstellen. Een regel zoals in afbeelding 7 maken we aan door eerst een “is greater than or equal to” predikaat naar het regel-deelvenster te slepen, en daarna het AankoopBedrag-fact links ervan te slepen, en het MinKortingBedrag rechts ervan te slepen. Deze conditie zal waar hetzij onwaar zijn.



Afbeelding 5. Schema van het U2U Order



Afbeelding 6. Configuratie van AankoopBedrag fact

Hierna is de actie aan de beurt. Deze actie wordt uitgevoerd als de conditie waar is. Om dit te definiëren slepen we het "Procent Korting"-fact naar het actiedeelvenster. De waarde die toegekend moet worden, kunnen we opnieuw uit het vocabularium slepen, of kunnen we direct ingeven. Het uiteindelijke resultaat zie je in afbeelding 7. Ook hier moeten we de policy publiceren alvorens deze te kunnen gebruiken. Hierna wordt de policy bruikbaar in een orchestratie.

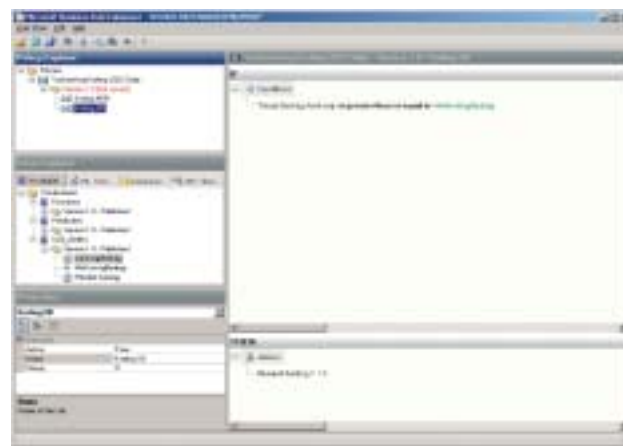
De Call Rules shape

Orchestraties bouwen we door verschillende shapes op een orchestratie te slepen. Hierbij vinden we shapes om XML-documenten te ontvangen, te verzenden, nieuwe XML-documenten aan te maken, enzovoort. De Call Rules-shape gebruiken we om policies mee op te roepen. Je vindt een voorbeeld van een Call Rule-shape in de orchestratie in afbeelding 1. Natuurlijk moeten we deze shape configureren en dat doen we met de "CallRules policy configuration dialog", getoond in afbeelding 8. Hierbij kunnen we uit een lijst met gepubliceerde policies kiezen, en daarna kiezen we de parameters die we wensen te gebruiken in de policy. De policy zal uitgevoerd worden op deze parameters.

In de hier gebruikte policy geven we een kopie van de ontvangen order door aan de policy die de korting voor de order invult. De order wordt dan doorgestuurd naar de volgende stap in het businessproces.



Afbeelding 8. Configuratie van de CallRules-shape



Afbeelding 7. Policy voor Toekenning U2U Korting

Business-rules oproepen vanuit andere applicaties

We mogen niet vergeten dat het Business Rules Framework eigenlijk een .NET class-bibliotheek is die los van BizTalk 2004 kan worden gebruikt. We kunnen onze businessregels dus ook hergebruiken in 'normale' applicaties, die we bouwen met Visual Studio .NET en een programmeertaal als VB.NET of C#. Een voorbeeld van de nodige code vind je in codevoorbeeld 1. Om dit te verwezenlijken moeten we eerst ons XmlDocument omvormen naar een TypedXmlDocument, wat eigenlijk een XmlDocument is met een extra naam. Deze naam wordt als factfinder gebruikt in de policy. Daarna maken we een instantie aan van een policy. We voeren de policy uit met de Execute-methode waar we de facts doorgeven als parameter(s). Een makkie voor zover we niet vergeten om een referentie te leggen naar Microsoft.RuleEngine.dll.

```
XmlDocument order = new XmlDocument();
order.Load( "InOrderKortingOk.xml" );
TypedXmlDocument typedOrder = new TypedXmlDocument( "RuleSchemas.Order", order );

Policy orderPolicy = new Policy("Toekenning Korting U2U Order");
orderPolicy.Execute( typedOrder );

order = (XmlDocument) typedOrder.Document;
order.Save( "OutOrderKorting.xml" );
```

Codevoorbeeld 1.

De Business Rules Engine oproepen in C#

Sterke combinatie

De Business Rules Engine vormt samen met BizTalk Server 2004 een sterke combinatie om onze business krachtig en flexibel te modelleren. Feit is dat je geen programmeur hoeft te zijn om mee te kunnen bouwen aan de uiteindelijke oplossing, of om de ontwikkelde oplossing te kunnen onderhouden. Eenvoudige deployment en opvolging van businessregels laten een efficiënte uitbating toe.

Referenties

msdn.microsoft.com/biztalk

MSDN over de Business Rules Engine:

www.msdn.microsoft.com/library/default.asp?url=/library/en-us/BTS2004IS/html/understanding_abstract_syfs.asp?frame=true

De BizTalk 2004 solution gebruikt in dit artikel met de schema's, orchestration en Console-project vindt je op www.microsoft.com/netherlands/msdn/netmagazine en www.u2u.net/articles

Peter Himschoot is Master in Computer Science, en werkt bij U2U als .NET-trainer en architect. U2U is als gecertificeerd .NET trainingscenter te Brussel gespecialiseerd in .NET-programmering (www.u2u.be). U2U verzorgt tal van MS BizTalk 2004-workshops in Europa en het Midden-Oosten. Peter en het U2U-team zijn te bereiken via: info@u2u.be

