

Services bouwen op basis van architectuur

WEB SERVICE SOFTWARE FACTORY ONDERSTEUNT BIJ ONTWIKKELEN VAN SERVICES

Ben je op zoek naar ondersteuning bij het ontwerpen van services? Zoek je een tool waardoor je sneller kunt ontwikkelen en die je het vervelende werk uit handen neemt? Ontdek wat Web Service Software Factory voor jou kan betekenen.

W e kennen het allemaal. We starten een nieuw project en één van de eerste projectactiviteiten is het bepalen van de optimale architectuur. Afhankelijk van de architect die bij het project betrokken is, resulteert dit in een bepaalde architectuur. Deze activiteit kost aanzienlijk wat tijd in de opstartfase van het project en het is maar de vraag of het resultaat voldoet aan de vooraf gestelde eisen. Daarbij komt, dat serviceoriëntatie steeds vaker een onderdeel is van de architectuureisen. Hoewel serviceoriëntatie een onderwerp is dat veel besproken wordt, blijkt het in de praktijk vaak lastig toe te passen. Samengevat, architectuur, en serviceoriëntatie in het bijzonder, resulteert in een aanzienlijk risico voor een project. Wordt het niet eens tijd voor een aantal 'best practices' op het gebied van serviceoriëntatie?

Nieuwe technologie, nieuwe hulpmiddelen

Met de komst van Visual Studio 2005, het bijbehorende .NET Framework 2.0 en het geplande Windows Communication Foundation hebben we alle elementen in handen om krachtige servicegeoriënteerde toepassingen te ontwikkelen. Echter, een technologie alleen is niet genoeg. We moeten natuurlijk wel weten hoe we deze technologie zo optimaal mogelijk gebruiken. Dit is de reden waarom Microsoft Patterns & Practices is gestart met een nieuw product met de officiële naam Web Service Software Factory. In de praktijk wordt deze naam afgekort tot Service Factory. De Service Factory is niet gewoon een nieuw building block zoals we dit kennen uit Enterprise Library. Het is een toolkit, bestaande uit richtlijnen, patterns, wizards en code snippets, die ons helpt bij het ontwerpen en bouwen van services, gebaseerd op Windows Communication Foundation of ASMX. Service Factory is gebaseerd op de Guidance Automation Toolkit en integreert naadloos in Visual Studio 2005. Hierdoor krijgen we een 'user experience' zoals we die nog niet eerder gezien hebben.

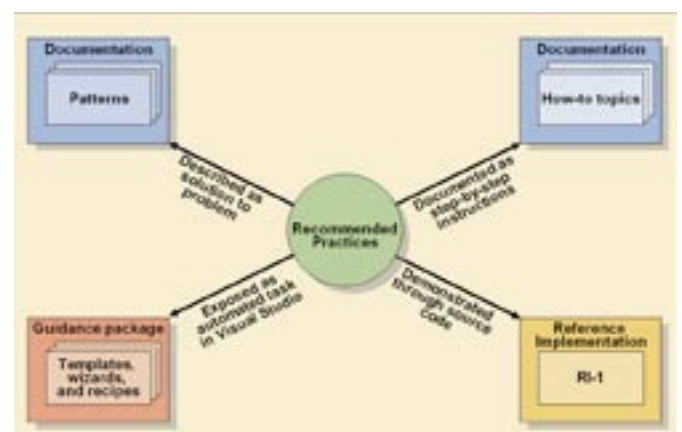
Waaruit bestaat Service Factory?

Service Factory is in de eerste plaats bedoeld voor architecten en lead developers. Deze kunnen Service Factory inzetten op hun project om zo een vliegende start te maken. Het gebruik hiervan ondersteunt de ontwikkelaars bij het uitvoeren van hun taken en stimuleert consistentie in het opgeleverde werk. Zoals we in afbeelding 1 kunnen zien, bestaat Service Factory uit verscheidene onderdelen die gezamenlijk de 'best practices' vertegenwoordigen voor het bouwen van servicegeoriënteerde toepassingen. Zo vinden we bijvoorbeeld uitgebreide 'how-to'-documenten die ons stap voor stap door het uitvoeren van veel voorkomende taken leiden.

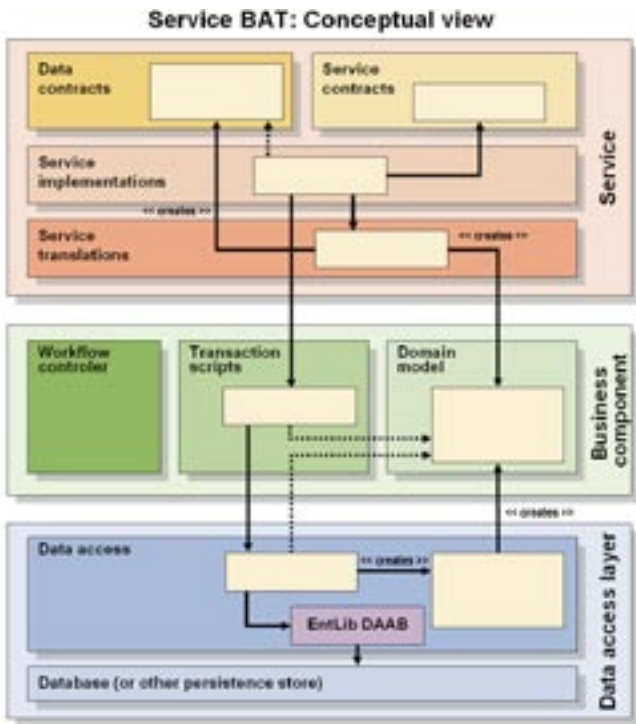
Daarnaast bevat Service Factory uitgebreide documentatie over het gebruik van design patterns. Problemen die we vaak tegenkomen in een servicegeoriënteerde toepassing worden opgelost met behulp van deze design patterns. Dit alles is natuurlijk fantastisch, maar eigenlijk niets nieuws. We hebben dit soort ondersteuning al eerder gezien in opleveringen zoals Enterprise Library. Wat wel nieuw is, is dat Service Factory deze 'best practices' beschikbaar stelt als 'Guidance Packages' gebaseerd op de Guidance Automation Toolkit. Deze 'Guidance Packages' integreren in Visual Studio en automatiseren vele activiteiten, die we normaal gesproken handmatig zouden uitvoeren. Uiteraard met inachtneming van de 'best practices'. Naast dit alles bevat Service Factory een zogenaamde 'reference implementation'. Dit is een realistisch, volledig uitgewerkt en werkend voorbeeld van een service die is ontwikkeld op basis van Service Factory. We kunnen dit voorbeeld gebruiken als referentie en/of studiemateriaal om te zien op welke manier Service Factory de verschillende onderdelen in onze service implementeert.

Voordelen van Service Factory

Door gebruik te maken van Service Factory verzekeren we ons van een snelle start van ons project. We weten dat we starten met een volwassen architectuur die gebaseerd is op 'best practices'. De wizards, code snippets en andere tools garanderen dat we ons houden aan de standaarden. Daarnaast krijgen we een hogere productiviteit. Een groot aantal handmatige taken wordt namelijk geautomatiseerd. Service Factory vertelt ons niet alleen uit welke



Afbeelding 1. Service Factory-onderdelen



Afbeelding 2. Service Factory conceptuele architectuur

componenten we onze services kunnen opbouwen en hoe we deze kunnen verdelen over de (conceptuele) lagen in onze architectuur, hij helpt ons ook bij het oplossen van de zogenaamde 'cross-cutting concerns'. Denk hierbij aan exception shielding, logging, versioning, security, enzovoort. Misschien wel het belangrijkste voordeel van Service Factory is echter de flexibiliteit en de aanpasbaarheid van het product. Service Factory dwingt ons bijvoorbeeld niet om activiteiten altijd in een bepaalde volgorde uit te voeren. We kunnen activiteiten weglaten en toevoegen en zelfs de hele structuur van Service Factory aanpassen. Als dit nog niet genoeg is, dan kunnen we natuurlijk ook de afzonderlijke onderdelen van Service Factory aanpassen, we krijgen immers alle source-code. Het is vooral deze flexibiliteit van Service Factory die hem een krachtig hulpmiddel maakt en er voor zorgt dat hij aansluit op de manier waarop wij graag willen werken.

Service Factory-architectuur

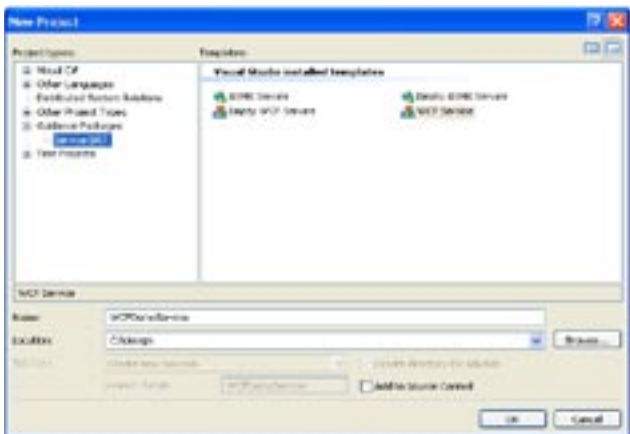
Nu we op hoofdlijnen weten wat Service Factory is, wordt het tijd om er wat gedetailleerder naar te kijken. Allereerst een kleine waarschuwing. Alle informatie en screenshots in dit artikel zijn gebaseerd op een 'early drop' van Service Factory. Tegen de tijd dat dit artikel gepubliceerd wordt, zijn er publicke versies beschikbaar die afwijken van de versie die gebruikt is voor dit artikel. In afbeelding 2 zie we een weergave van de

conceptuele architectuur zoals die door Service Factory gehanteerd wordt. In dit overzicht zie je dat er is gekozen voor een duidelijke scheiding van verantwoordelijkheden. Zo zie je bijvoorbeeld een splitsing tussen de definitie van het servicecontract, datacontract en service-implementatie. Verder zie je dat er een onderscheid gemaakt wordt tussen 'Service'-, 'Business'- en 'Data Access'-gerelateerde componenten. Je ziet ook dat transformaties tussen het servicedomein en het businessdomein worden geregeld in een aparte logische component, in dit geval 'Service Translations'. Helaas kunnen we in dit artikel niet dieper ingaan op de achterliggende reden van deze en andere opdelingen in de conceptuele architectuur. We zullen zo zien dat deze opdeling niet alleen een mooi plaatje is in een helpfile, maar dat Service Factory ons ook ondersteunt bij het implementeren van deze conceptuele architectuur.

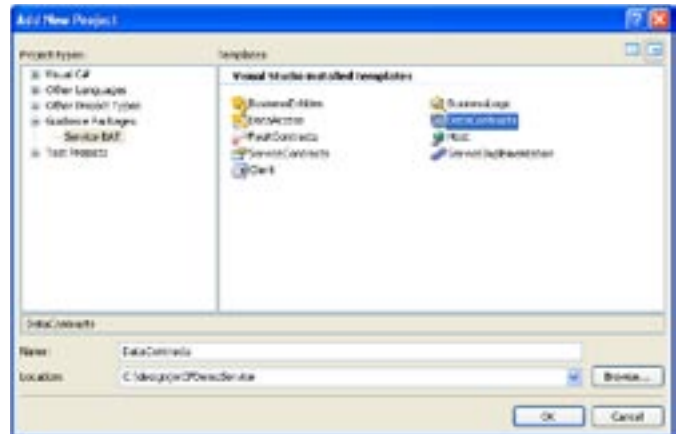
Service Factory solution-structuur

Zoals gezegd integreert Service Factory volledig in de Visual Studio 2005-omgeving. Dit betekent dat we alle acties en ondersteuning van Service Factory beschikbaar hebben in onze ontwikkelomgeving. Laten we eens kijken hoe dit werkt. Om een service te ontwikkelen met behulp van Service Factory zullen we allereerst een Visual Studio-solution aan moeten maken. Service Factory stelt hiervoor templates beschikbaar die we kunnen gebruiken als basis voor onze service. In afbeelding 3 kunnen we zien dat er een Service BAT-item ('code name' van het project) is toegevoegd onder 'Guidance Packages'. Je ziet dat Service Factory twee templates beschikbaar stelt voor ASMX en twee voor Windows Communication Foundation. Service Factory ondersteunt dus beide technologieën. We zien hier ook meteen een stuk van de flexibiliteit van Service Factory. Als we kiezen voor de WCF Service-template, dan genereert Service Factory een volledige Visual Studio solution-structuur voor ons. Hierin zien we bijvoorbeeld een apart project voor alle verschillende onderdelen in de Service Factory-architectuur van afbeelding 2. Ieder project in de solution-structuur vertegenwoordigt een aantal onderdelen uit deze architectuur. Vaak willen we echter zelf wat meer invloed uit oefenen op de solution-structuur. We willen bijvoorbeeld zelf bepalen welke projecten we nodig hebben binnen onze solution-structuur. Ook dit is mogelijk. We kiezen in dit geval voor de 'Empty WCF Service'-template. Service Factory genereert nu een lege solution-structuur die we vervolgens zelf kunnen indelen op basis van onze requirements.

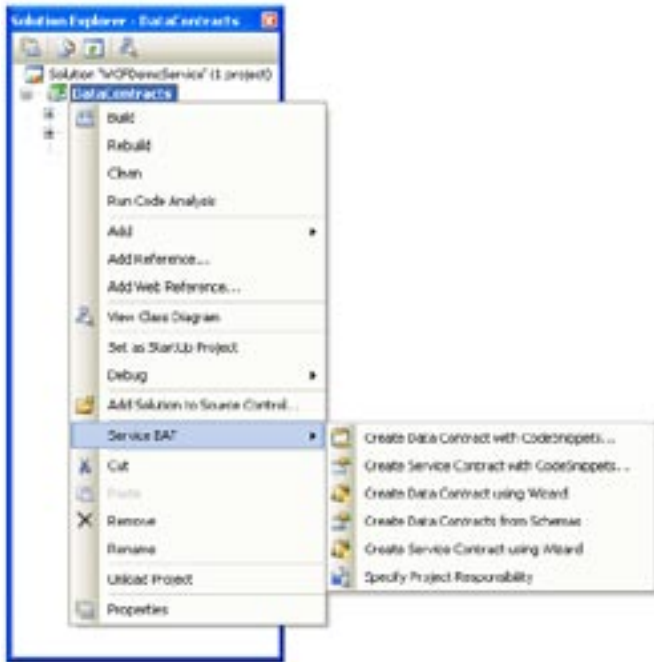
In afbeelding 4 zien we dat Service Factory verschillende projecttypen beschikbaar stelt. We kunnen bijvoorbeeld een 'DataContract'-project toevoegen aan onze lege solution. Als we dit doen krijgen we niet alleen een nieuw project, maar ook alle bij dat projecttype behorende tools en wizards tot onze beschikking. Deze tools kunnen we vervolgens gebruiken bij het ontwikkelen van de componenten die horen bij dat specifieke projecttype.



Afbeelding 3. Service Factory-templates



Afbeelding 4. Service Factory-projecttypen



Afbeelding 5. Beschikbare acties voor aangepast datacontractproject.

Service Factory biedt nog meer flexibiliteit. Zoals we gezien hebben, wordt op basis van de verantwoordelijkheden van een projecttype bepaald welke acties een project in onze solution ondersteunt. Een 'DataContract'-project zal bijvoorbeeld acties ondersteunen die gerelateerd zijn aan het maken van een datacontract. Een 'ServiceContract'-project biedt ondersteuning bij het maken van de service-interface, enzovoort. Ook dit kunnen we aanpassen. We kunnen verantwoordelijkheden toekennen aan een projecttype. In sommige situaties is het namelijk helemaal niet wenselijk om voor elk onderdeel van onze service een apart project te hebben in onze Visual Studio solution. In afbeelding 5 kunnen we zien dat we het 'DataContract'-projecttype dusdanig hebben aangepast dat we ook de beschikking hebben over de 'ServiceContract'-gerelateerde acties. Kortom, ook op dit vlak kunnen we Service Factory optimaliseren.

Ontwikkelen met Service Factory

Nadat we met behulp van Service Factory een solution-structuur voor onze service gegenereerd hebben, kunnen we starten met het bouwen van onze service. We hebben net al gezien dat het maken van de datacontracten voor onze service één van de activiteiten is die we moeten uitvoeren. Zonder diep op het concept datacontract in te gaan, kunnen we zeggen dat we in een datacontract beschrijven welke data noodzakelijk zijn voor het uitvoeren van een serviceoperatie. Een datacontract dient als invoer of uitvoer van een serviceoperatie. Voor dit artikel gaat het te ver om dit concept in detail te behandelen. Uiteraard kan je hier in Service Factory veel meer informatie over vinden. Waar het nu om gaat is dat we begrijpen dat Service Factory ons helpt bij het maken van deze datacontracten. Als we nog even terugkijken naar afbeelding 5 dan zien we dat Service Factory verschillende manieren ondersteunt om een datacontract te maken. We kunnen kiezen uit het maken van een datacontract op basis van een wizard, code snippet of XSD-schema. Door het uitvoeren van de verschillende acties die in Service Factory beschikbaar zijn, kunnen we stap voor stap de onderdelen van onze service bouwen. Service Factory biedt daarbij de mogelijkheden zelf te bepalen welke tool of wizard het beste aansluit.

Een blik in de toekomst

We hebben nu een aantal features van Service Factory voorbij zien komen. Maar er is meer. Service Factory biedt bijvoorbeeld een strategie voor het afhandelen van excepties die optreden in onze services. Het biedt een tool die ons helpt bij het mappen van businessobjecten op datacontracten. Het helpt ons met het oplossen

van security-problemen. Kortom, Service Factory begeleidt ons met alles dat valt tussen de service-proxy en de database. Service Factory biedt ons ook een blik in de toekomst voor wat betreft het Software Factories-initiatief van Microsoft. Hoewel Service Factory nog niet voldoet aan alle eisen van het concept Software Factory, zien we duidelijk welke richting het de komende tijd op zal gaan. Voor nu is het belangrijk te onthouden dat Service Factory ons helpt bij het ontwikkelen van services. Service Factory ondersteunt Windows Communication Foundation en ASMX en integreert met Visual Studio.NET 2005. Service Factory levert een architectuur die gebaseerd is op 'best practices'. Twee van de belangrijkste uitgangspunten van Service Factory zijn aanpasbaarheid en flexibiliteit. Service Factory heeft een eigen community-site. Als je interesse hebt in bovenstaande, meld je dan aan bij de Service Factory-community. Download een preview van Service Factory en geef je feedback. Voor iedereen die interesse heeft in architectuur, services, Windows Communication Foundation en ASMX is dit zeker de moeite waard.

Edward Bakker is werkzaam als softwarearchitect bij LogicaCMG. Je kunt hem bereiken via Edward.Bakker@logicaCMG.com of www.EdwardBakker.nl

Referenties:

<http://practices.gotdotnet.com/projects/svcfactory>
<http://msdn.microsoft.com/practices/>
<http://msdn.microsoft.com/practices/topics/appint/default.aspx>
<http://msdn.microsoft.com/vstudio/teamsystem/workshop/gat/default.aspx>
<http://msdn.microsoft.com/vstudio/teamsystem/workshop/sf/default.aspx>

(advertentie Microsoft Press)



The Security Development Lifecycle
 ISBN: 0-7356-2214-0
 Auteur: Michael Howard and Steve Lipner
 Pagina's: 352