

Met WinFX veranderen de mogelijkheden van applicaties en de manier waarop deze ontwikkeld zullen worden: hoe applicaties eruit zien en beleefd worden, hoe je met data omgaat en vooral, hoe je een applicatie van binnen structureert en met de buitenwereld laat communiceren. WinFX brengt fundamentele wijzigingen voor ontwikkelaars en gebruikers. In dit artikel wordt er een aantal besproken aan de hand van een nieuwe applicatie voor een technische dienst.

thema

WinFX: communicatie en workflow

Meer functionaliteit in minder tijd

De gewenste functionaliteit bestaat uit het kunnen invoeren, bijhouden en toewijzen van opdrachten en mogelijk aanpassen van het proces. Bovendien moeten gebruikers tijdens het proces geïnformeerd blijven en diverse overzichten kunnen raadplegen. Dat klinkt mogelijk niet als een hightech applicatie, maar toch is dit een typisch scenario om gebruik te maken van WinFX.

IN HET KORT Later dit jaar komt er een nieuwe versie van het Windows-besturingssysteem: Windows Vista. Deze versie, in server- en client-varianten, heeft WinFX geïntegreerd. WinFX komt ook beschikbaar voor Windows Server 2003 en Windows XP. WinFX is een gratis framework en een logische aanvulling op .NET Framework 2.0. Het bestaat uit drie onderdelen: Windows Presentation Foundation (WPF), Windows Workflow (WF) Foundation en Windows Communication Foundation (WCF). Windows Presentation Foundation is de basis van indrukwekkende vector graphics voor smart clients. Meer hulpmiddelen komen beschikbaar, waaronder grafische programma's als Graphic, Interactive en Web Designer van de Microsoft Expression Suite. Daarnaast zijn er al conversieprogramma's om bestaande tekeningen van bijvoorbeeld Adobe Illustrator om te zetten naar de bestandsindeling voor WPF. Voor webapplicaties zijn er diverse voorbeelden van samenwerking van Atlas, de Microsoft-webtechniek gebaseerd op AJAX, met WCF. In dit artikel ligt de nadruk echter op workflow en communicatie.

WORKFLOW EENVOUDIGER IN EEN APPLICATIE

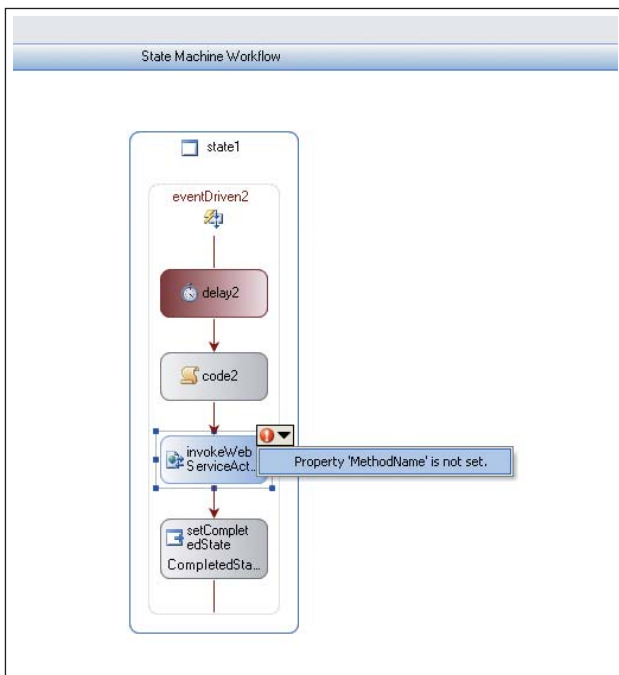
Workflow was lang het terrein van kostbare, weinig flexibele pakketten. Als alternatief bouwden veel organisaties één of meerdere maatwerkoplossingen voor workflow. Microsoft deed dit zelf ook. Tijdens PDC05 werd bekend dat daar verandering in komt. Toen kondigde Microsoft Windows Workflow Foundation aan om workflow op een uniforme manier in producten als Sharepoint en toekomstige versies van CRM en BizTalk toe te passen. Er zijn zelfs plannen om WF aan te bieden als mechanisme voor pageflow in applicaties. Bovendien is WF door Microsoft's partners en klanten uit te breiden of in maatwerkoplossingen toe te passen.

WF is precies wat de naam zegt: een fundament of een basis. Er is gezorgd voor een snelle start voor ontwikkelaars door het aanbieden van opslag- en analyse-mogelijkheden van workflow-instanties met behulp van een database en het aanbieden van veel voorbeelden. In WF is echter een aantal zaken niet geregeld. Het is niet een kant-en-klaar product dat je kunt installeren en direct aan eindgebruikers kunt geven. Je moet het toepassen. Als je dat wilt, kun je met WF wel alle kanten op. Zo is er ondersteuning voor transacties, compenserende acties, voor het tijdelijk opslaan en later reactiveren.

Microsoft ziet workflow als een verzameling activiteiten vormgegeven in een aaneengeschakeld of reactief patroon, een sequentiële of een state-machine workflow. Een voorbeeld van standaard beschikbare activiteiten zijn IfElse, While, Parallel en InvokeWorkflow. Nu is

eigenlijk iedere if-else constructie in een workflow te vatten, maar dat is niet altijd nodig. Veel logica-constructies mogen gecodeerd worden omdat ze altijd hetzelfde moeten functioneren binnen een proces. Wanneer een business- of beheerafdeling het proces echter wil bepalen of het inzichtelijk wil hebben, dan is Windows Workflow Foundation een goede optie.

WORKFLOW ONTWIKKEL- EN IMPLEMENTATIE-OMGEVING Microsoft Visual Studio 2005 is de omgeving waarin workflows kunnen worden ontwikkeld. Zie de workflow als een form en de activiteiten als controls op dat form, vormgegeven in een domeinspecifieke taal. Je start met sequential of state-machine workflow project in een console-applicatie of een library. Vervolgens kun je hier meerdere typen workflows en activiteiten, wat beide gewoon classes zijn, aan toevoegen. De workflow kan in de designer worden gedefinieerd. De designer is herkenbaar voor BizTalk gebruikers en laat zich eenvoudig bedienen.



AFBEELDING 1. In Visual Studio 2005 is volop ondersteuning voor het ontwikkelen en debuggen met WF: Na het toevoegen van een InvokeWebservice activiteit en het instellen van het adres, geeft de service aan dat er nog een methode moet worden gespecificeerd.

De designer is een herbruikbaar component dat door ontwikkelaars kan worden aangeboden om workflow in maatwerkapplicaties te definiëren, door het gebruiken van DesignSurface en WorkflowDesignerLoader uit de System.Workflow.ComponentModel.Design namespace. Workflows kunnen worden opgeslagen in een specifiek XML-formaat, waardoor ook op andere manieren, bijvoorbeeld met een wizard zoals in

SharePoint 2007, workflows kunnen worden samengesteld en door WF geïnterpreteerd. Zowel de XML-definities als de codebestanden kunnen worden geïnterpreteerd door Windows Workflow Compiler (wfc.exe). Om workflows uit te voeren is er een host en een runtime nodig. De host kan de applicatie zijn waarin de workflow wordt gebruikt. De runtime starten kan door een instantie te maken van de runtime en vervolgens een instantie van de gedefinieerde workflow te creëren.

```
WorkflowRuntime workflowRuntime = new
WorkflowRuntime();
WorkflowInstance instance =
    workflowRuntime.CreateWorkflow
        (typeof(Workflow1));
instance.Start();
```

ACTIVITEITEN VORMEN DE WORKFLOW Als activiteiten een workflow vormen, dan moeten activiteiten flexibel aanpasbaar en uitbreidbaar zijn. Dankzij de System.Workflow.ComponentModel.Activity class zijn ze dat ook. Een eenvoudige activiteit maak je al door een class hiervan te laten erven en de Execute methode te implementeren, die een ActivityExecutionContext ontvangt en een ActivityExecutionStatus retourneert. Mogelijke statuswaarden zijn initialized, executing, cancelling, faulting, compensating en closed. Een handige uitbreiding kan bijvoorbeeld een SendMailActiviteit zijn, die e-mails verstuurt naar aanleiding van een statusovergang tijdens de workflow.

Windows Workflow Foundation kent meer standaardactiviteiten, waaronder InvokeWebservice en Code. Deze laatste twee zijn uitermate geschikt om met externe systemen te communiceren. De ultieme communicatiepartner is Windows Communication Foundation. Hiervoor zijn al specifieke activity generators ontwikkeld, waarmee WCF-instanties kunnen worden aangesproken. Bovendien kan een workflow zich aanbieden als webservice door de WebServiceInput activiteit.

COMMUNICEREN MET WEBSERVICES Uitgangspunt voor het ontwikkelen van Windows Communication Foundation was het doorontwikkelen van service-oriëntatie, maar dan op een manier die beter presteert, gemakkelijker met andere platformen communiceert, eenvoudiger te beheren en meer uitbreidbaar is. Service-oriëntatie is een aanpak die veel bedrijven en architecten onderschrijven, maar aan de huidige technieken zitten nogal wat uitdagingen. .NET Framework 2.0 en Web Services Enhancements 3.0 doen een stap in de goede richting op het gebied van uitbreidbaarheid en interoperabiliteit. WCF is de verzameling van technie-

ken als webservices, webservice-uitbreidingen, messaging, remoting en COM+ in een uniforme structuur met de meeste voordelen van deze technieken.

Een ander uitgangspunt was dat transport- en netwerkselectie configureerbaar moest worden en dat bedrijfslogica gescheiden moest zijn van technische implementatie. Bovendien is voor interoperabiliteit ondersteuning van WS-* standaarden onmisbaar, zoals beveiliging, betrouwbaar communiceren en transacties. WCF voldoet aan deze eisen en implementeert ze op een toegankelijke manier. De gebruikelijke aanpak voor het ontwikkelen van een WCF-service is dat je een contract definieert, implementeert en op een bepaalde manier bindt aan een adres. In het contract wordt beschreven welke gegevens en methodes kunnen worden aangesproken, in de binding wordt beschreven op welke manier gecommuniceerd kan worden en het adres beschrijft de locatie waar de service kan worden benaderd. Eén van de principes van serviceoriëntatie is dat grenzen expliciet zijn. WCF houdt zich hieraan en laat pas een type en methode zien als dat wordt aangegeven. Dat kan je doen op de volgende manier.

EEN WCF-SERVICE Maak een nieuw project en leg een referentie naar `System.ServiceModel`. Maak een interface met het `ServiceContract` attribuut en definieer daarin methodes met het `OperationContract` attribuut. In een migratietraject van eerdere technieken naar WCF zou je ervoor kunnen kiezen om de contractattributen direct op de serviceclass te definiëren in plaats van een interface onttrekken en deze van attributen te voorzien.

```
[ServiceContract]
public interface ITDService {
    [OperationContract]
    int Process(TDJobMessage jobmessage);
}
```

Een `ServiceContract` kan onder andere worden voorzien van parameters als configuratienaam, beschermingsniveau en sessie. In een `OperationContract` kun je met name het communicatiepatroon aangeven, of het eenrichtingsverkeer is en ook het beschermingsniveau. Implementeer de interface in een class en geef inhoud aan de methode. Vervolgens zijn er gegevens die je heen en weer wilt sturen. Deze gegevens in de vorm van een business-object, kun je definiëren in `DataContract`. De gegevens zijn zo gescheiden van de implementatie van bedrijfsoperaties. Maak een class met het `DataContract` en het `Serializable` attribuut. Pas de `DataContract`-en `DataMember`-attributen toe op respectievelijk de business class en de betreffende property's.

Door gebruik te maken van deze attributen en hun

parameters kun je de interface beïnvloeden. Zo kan je een namespace opgeven en de volgorde van de `DataMembers`. Wil je meer controle over de interface, dan kun je ook gebruik maken van het `MessageContract` attribuut. Hiermee kun je naast het datacontract specifiek aangeven hoe het bericht eruit moet zien opgesplitst in `MessageHeader` en `MessageBody`.

```
[MessageContract]
public class TDJobMessage{
    [MessageHeader]
    public string[] UserPass;
    [MessageBody]
    public TDJob TDJob; }
}
```

Vervolgens moet de service draaien in een host. Je kunt de service hosten in een applicatie (windows service) door een instantie van `System.ServiceModel.ServiceHost` te maken, maar je kan ook Microsoft Internet Information Services (IIS) met ASP.NET 2.0 gebruiken zoals hieronder. Eerst maak je een .svc bestand waarin de verwijzing staat naar `ServiceHost` en `Assembly`:

```
<%@ServiceHost language="C#" Debug="true"
service="SRService.TDService" %>
<%@Assembly Name="SRService" %>
```

Vervolgens maak je een web.config bestand waarin de configuratie van de service staat.

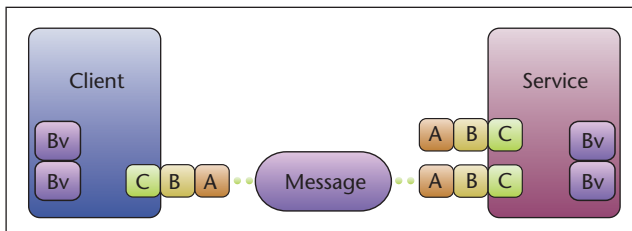
```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.serviceModel>
    <services>
      <service name="SRService.TDService">
        <endpoint contract="SRService.ITDService" binding="wsHttpBinding"/>
      </service>
    </services>
    <behaviors>
      <behavior name="returnFaults" returnUnknownExceptionsAsFaults="true" />
    </behaviors>
  </system.serviceModel>
</configuration>
```

Nu maak je in IIS een virtuele directory aan, die verwijst naar de locatie van het .svc bestand. Let er op dat op diezelfde locatie een bin folder bestaat met daarin de dll van de service. De service is klaar en nu is het tijd om een client te maken en die te verbinden met de service.

De Microsoft Service Model Metadata Tool, svcutil.exe, is hierbij een handig hulpmiddel.

```
C:\Program Files\Microsoft SDKs\Windows\v1.0\  
bin\SvcUtil.exe http://localhost/SRService/  
TDSservice.svc?WSDL
```

Dit commando levert, zonder gebruik van de vele switches, een .cs en een .config bestand op. Hiermee kan de client de service vinden en de operaties aanspreken. Het codebestand bevat definities van de service en een proxy om tegen te programmeren. Het configuratiebestand bevat een endpoint, de ingang van de service, samengesteld uit een address, een binding en een contract.



AFBEELDING 2. Service, client en address, binding en contract.

ADRES BINDING CONTRACT Dit is het ABC van Windows Communication Foundation. Het adres geeft aan waar de service zich bevindt in de vorm van een Uniform Resource Identifier (URI). In dit adres kan protocol en poort worden aangegeven. De binding geeft aan hoe het contract aan het adres wordt gebonden. Hierin worden zaken geregeld als transport, beveiliging en codering afhankelijk van het betreffende protocol. De beschikbare bindings zijn basicHttp-, msmqIntegration-, netMsmq-, netNamedPipe-, netPeerTcp-, netTcp-, wsDualHttp-, wsFederationHttp-, wsHttpBinding. De http-varianten zijn interoperabel en de anderen kunnen van WCF naar WCF gebruikt worden.

WCF kent service-, data- en message-contracten. Een service-contract gebruik je altijd en beschrijft de methodes in de vorm van WSDL. Een data-contract kan je gebruiken als je complexe data en types wilt uitwisselen in de vorm van XSD. Eerder noemde ik ook het message-contract waarmee je het uitgewisselde bericht beïnvloedt in de vorm van SOAP.

Naast het ABC van endpoints is er nog een belangrijke manier om services te definiëren en dat is door het gebruik van behaviors. Behaviors zijn gedragingen van een service of van een operatie in respectievelijk een ServiceBehavior en een OperationBehavior. Met deze gedragingen kun je zaken beïnvloeden als instan-



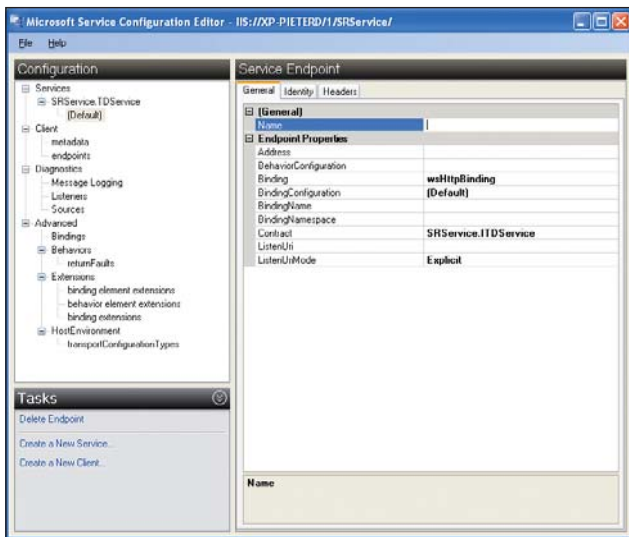
Bent u ICT-professional of ICT-Decision maker?

Computable is de meest gelezen informatiebron voor ICT-professionals en ICT-managers in Nederland. Met altijd het belangrijkste nieuws, objectieve scherpe analyses, vacatures en onafhankelijke productinformatie. Bent u als ICT'er werkzaam, dan heeft u recht op een kosteloos abonnement.

Ga nu naar www.abonneren.nl/computable en meld u aan!

Computable, dé ICT-informatiebron

Computable



AFBEELDING 3. WCF eigenschappen en instelling zet je heel eenvoudig met behulp van de Service Configuration Editor.

tiëren, transacties, identiteit en het maximale aantal en gelijktijdige verbindingen. Enkele van deze eigenschappen waren voorheen instelbaar in COM+ of EnterpriseServices, maar met WCF kunnen ook flexibele maatwerkgedragingen ontwikkeld worden. De ontwikkelaar voegt in de client referenties toe naar `System.Runtime.Serialization` en `System.ServiceModel`, programmeert code tegen de proxy aan en de applicatie kan getest worden.

De fundamenteën van WinFX zijn nog volop in ontwikkeling. De ondersteuning voor het maken van workflow gaat op dit moment het verste. Voor WCF zijn er wel tools die het leven van een communicatie-ontwikkelaar makkelijker maken: `svconfigeditor.exe` (Microsoft Service Configuration Editor) en `svctraceviewer.exe` (Microsoft Service Trace Viewer).

De editor helpt bij het maken van configuratiebestanden, proxy's en instellingen. Ook voorziet hij je van informatie voor het verhelpen van problemen met een service. Microsoft Service Trace Viewer is specifiek gericht op het analyseren en verhelpen van problemen door het bekijken, groeperen, en filteren van berichten.

STRUCTUUR IN DE GROTE LIJNEN Windows Workflow Foundation en Windows Communication Foundation van WinFX vormen het terrein van de ontwikkelaar van bedrijfslogica. Door het gebruik van workflow in een applicatie komt er onderhoudbaarheid en structuur in flows en is het inzichtelijk voor de belanghebbenden. Communicatie op de WCF-manier helpt ontwikkelaars om instelbare, servicegeoriënteerde applicaties en lagen te bouwen die beter presteren en interoperabel zijn. De combinatie van Workflow en

Communication Foundation is extra sterk doordat WF goed gebruikt kan worden voor het coördineren van werk voor verschillende services. Tenslotte biedt Workflow helder inzicht in het asynchroon en gelijktijdig aanroepen van services.

Motivatie

Windows Workflow Foundation:

- * Inzichtelijkheid van statussen door WorkflowTracking
- * Informeren van gebruikers door SendMailActivity,
- * Aanpasbare workflow door designer in de applicatie

Windows Communicatie Foundation:

- * Mogelijkheid om efficiënt en beveiligd te communiceren,
- * Mogelijk voor aanbieden van functionaliteit aan andere applicaties,

In dit artikel is gebruik gemaakt van Microsoft WinFX Community Technology Preview Februari 2006.

Zelf aan de slag met WinFX?

- * Visual Studio 2005
- * WinFX Runtime Components
- * Windows SDK
- * Visual Studio Development Tools for WinFX
- * Visual Studio Extensions for Windows Workflow Foundation

Referenties

- <http://msdn.microsoft.com/winfx>
- <http://windowscommunication.net>
- <http://www.windowsworkflow.net>
- <http://www.avanade.nl>

Pieter de Bruin is consultant bij Avanade, een joint-venture van Accenture en Microsoft. (pieterd@avanade.com)