

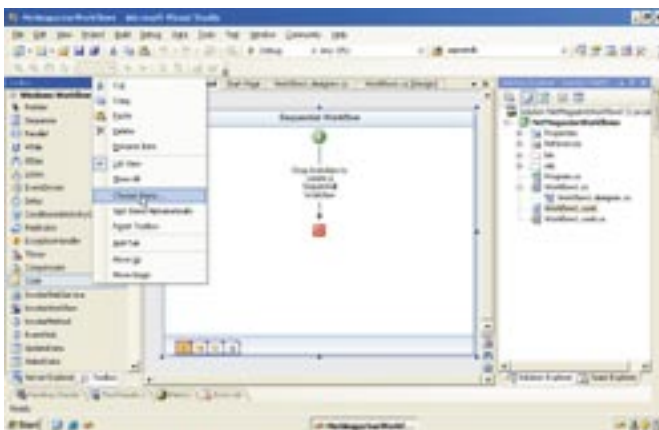
# De kracht van workflow

## WINDOWS WORKFLOW FOUNDATION

Hoe vaak zijn we niet bezig om in onze applicatie een vorm van workflow te implementeren? Workflow zien we terug in vrijwel ieder businessproces dat we ondersteunen met software. Ook veel producten van Microsoft kennen vormen van workflow (SharePoint, BizTalk, ASP.NET, CRM...) In dit artikel wil ik laten zien hoe Microsoft hiervoor platformondersteuning gaat uitbrengen waar wij onze applicaties mee kunnen ontwikkelen.

Op de Microsoft Professional Developer Conference (PDC 05) in september van dit jaar is Windows Workflow Foundation (codenaam: WinOE) aangekondigd. Feitelijk is het project drie jaar lang buiten de publiciteit gehouden en wist buiten Microsoft, op enkele partners na, niemand van het bestaan van het project. Windows Workflow Foundation wordt uitgebracht als onderdeel van WinFX, een framework bovenop het Microsoft .NET Framework dat in essentie de Windows API gaat vervangen en ons een nieuwe Managed API biedt voor het ontwikkelen van onze applicaties. Op de MSDN-site van Microsoft zijn al diverse artikelen verschenen over de concepten van Windows Workflow Foundation. Ik ga daarom niet al te diep in op de onderliggende architectuur of het waarom van Windows Workflow Foundation. Het doel van dit artikel is om je belangstelling te wekken, zodat jij je meer gaat verdiepen in de informatie die Microsoft biedt. In de referenties is een aantal nuttig links te vinden. Vooral de hands-on labs zijn een goed startpunt. Als we aan workflow denken, dan denken we aan een aantal stappen, ook wel activiteiten genoemd. De stappen kunnen vaak door software uitgevoerd worden, maar er zijn ook menselijke stappen in een bedrijfsproces. Windows Workflow Foundation ondersteunt beide, zowel voor korte synchrone calls als voor langdurende transacties.

Zoals te zien is in afbeelding 1 lijkt de Workflow-designer erg veel op de Orchestration Designer zoals we deze kennen van BizTalk. Hoewel Microsoft wel degelijk inspiratie heeft opgedaan voor de Orchestration Designer is het een volledig nieuwe desig-



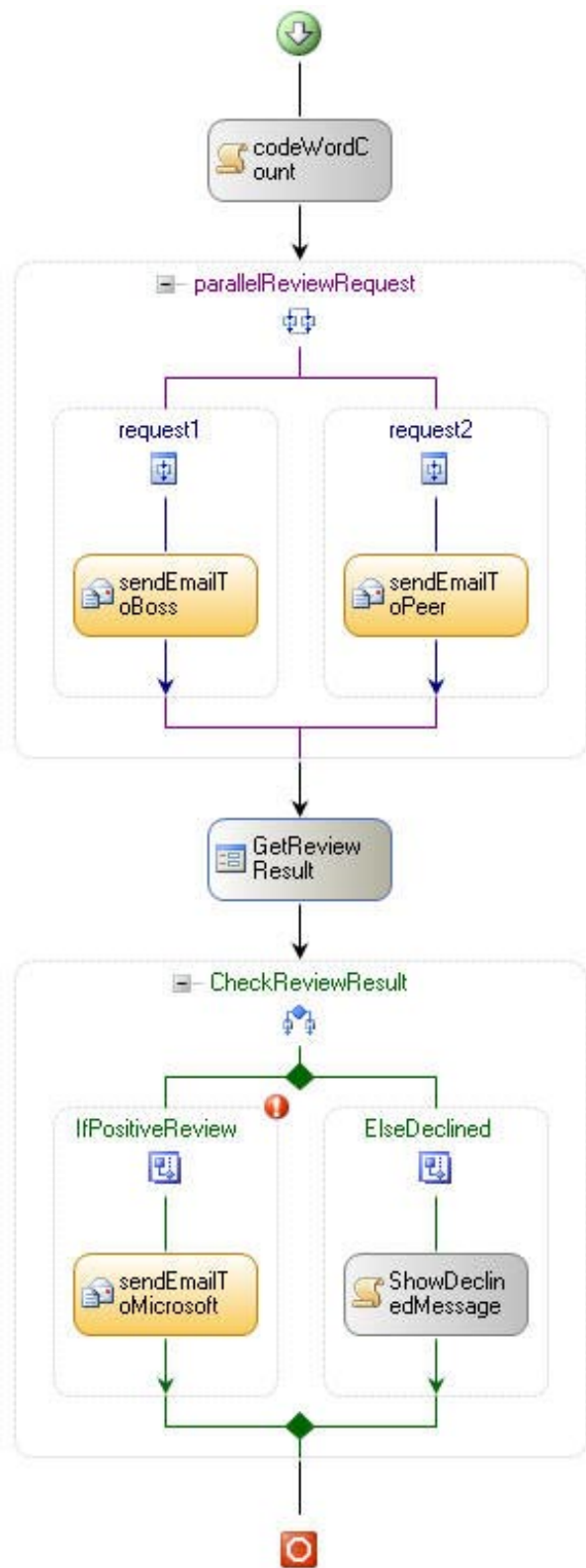
Afbeelding 1. De Workflow-designer

ner waarop, vanaf de volgende versie van BizTalk (na BizTalk Server 2006), ook de Orchestration engine gebaseerd zal worden. Net als de vele andere designers in Visual Studio is het mogelijk ook deze designer buiten Visual Studio te 'hosten', zodat de kracht ervan ook in je eigen applicaties gebruikt kan worden. Ook zal Office "12", inclusief SharePoint, veelvuldig gebruik maken van de kracht van Windows Workflow Foundation. In afbeelding 1 zijn diverse activiteiten te zien in de toolbox van Visual Studio. Een aantal activiteiten wordt standaard meegeleverd, andere bouw je zelf of kan je kopen als onderdeel van Microsoft- en third party-producten. Grofweg kent Windows Workflow Foundation twee typen workflow, de Sequential Workflow, die ik in dit artikel gebruik en de State Machine Workflow. De Sequential Workflow beschrijft grofweg een aantal opeenvolgende stappen, waarbij de State Machine Workflow een stuk dynamischer is en er ook teruggegaan kan worden naar bepaalde states. State Machine Workflow zal je vaak toepassen als er (meer) mensen betrokken zijn bij het proces en de flows nogal dynamisch van aard zijn. Een ander voorbeeld van wat goed te modelleren is in een State Machine Workflow is de flow van applicatieschermen in bijvoorbeeld een ASP.NET-applicatie. Voor degenen die zich al eens hebben afgevraagd waarom in Enterprise Library geen User Interface Process Block te vinden was, is hier het antwoord. Workflow biedt een elegantere oplossing.

### Eerste workflow-applicatie

In dit artikel ga ik een workflow-applicatie maken. Een applicatie voor het goedkeuren en reviewen van een artikel in .NET Magazine. Zoals niet iedereen zal weten, is het aanleveren van een artikel voor .NET Magazine een hele kluit. Plaatjes, tekst, en wat dies meer zij moeten allemaal aan voorwaarden voldoen en je mag niet zonder meer een oneindige hoeveelheid tekst produceren. Bovendien is het van belang dat het artikel interessant genoeg is voor de professionele ontwikkelaar. Veel van de auteurs van artikelen in .NET Magazine gebruiken collega's om het artikel initieel te laten reviewen. Uiteraard wordt een artikel daarnaast ook door de redactie gelezen voordat het uiteindelijk wordt opgemaakt en aan-geleverd aan de drukker. Dit proces (het eerste idee tot en met het drukken en bezorgen) kan wel tot drie maanden duren. We gaan een deel van dit proces automatiseren.

De belangrijkste criteria voor het aanleveren van een artikel is dat het een normaal ASCII- tekstbestand is met de tekst en dat verder alle illustraties en codevoorbeelden in aparte bestanden staan. Uiteraard moet de tekst wel netjes verwijzen naar de



Afbeelding 2. De workflow van het reviewproces

(genummerde) codevoorbeelden en illustraties. Verder is er een maximum van ongeveer 2000 woorden per artikel, afhankelijk van het aantal pagina's dat toebedeeld is. Plaatjes en codevoorbeelden kosten weer woorden. Indien het artikel niet voldoet aan de criteria zal de redactie het artikel terugsturen met het nodige commentaar en moet het artikel door de auteur aangepast worden om het vervolgens zo spoedig mogelijk terug te sturen, want de planning is strak. Te laat is te laat en dan wordt je artikel pas drie maanden later gepubliceerd. In afbeelding 2 is een overzicht te zien van de applicatie die we gaan bouwen. Dit overzicht is gemaakt met de Workflow Designer en biedt een leesbaar overzicht van de workflow die ook voor je klant erg



Afbeelding 3. Het reviewformulier

herkenbaar zal zijn. De code die bij de workflow hoort staat in een zogenaamde code beside file.

Voor het gemak maken we eerst een Windows Form met drie tekstboxen. De titel, subtitel, schrijver en de tekst. Het formulier staat afgebeeld in afbeelding 3. De code die bij het formulier hoort, laat ik vanwege ruimtegebrek achterwege, maar is te downloaden vanaf de .NET Magazine-site of mijn blog (net als de rest van de applicatie uit dit artikel). Na het drukken op de knop 'Verstuur' gaan we de workflow starten en gaat het proces lopen.

In het diagram in afbeelding 2 heb je al gezien wat de eerste stap is. We gaan een wordcount doen van het artikel. Hiervoor gebruiken we de code-activity waarin ik een grof wordcount-algoritme heb geïmplementeerd, zie codevoorbeeld 1. In de Workflow Designer kan je op de code-activity dubbelklikken om je code in te voeren, net als we dat gewend zijn van bijvoorbeeld de winforms designer. De designer genereert zelf de code die zorgt voor de aanroep van deze handler. De output wordt bewaard in de member-variabele van de Windows Workflow Foundation, genaamd amountOfWords. Vervolgens versturen we het artikel naar twee collega's. Nu hebben we direct een probleem, want de MailActivity bestaat standaard niet en moet je dus zelf maken, ware het niet dat Microsoft op [www.windowworkflow.net](http://www.windowworkflow.net) een sample activity heeft staan die precies doet wat ik wil en deze is zelfs zo geschreven dat hij er ook mooi uitziet in de Workflow Designer, zoals je kunt zien in afbeelding 2. Om de mails te versturen gebruiken we de Parallel Activity; waarom immers de calls na elkaar doen (velen van ons hadden dit in gewone code wel gedaan of het met een BCC opgelost). Dit principe kan je bijvoorbeeld ook gebruiken om meer calls naar een aantal externe systemen uit te voeren in plaats van deze één voor één te doen, zoals op dit moment in veel applicaties gebeurt. De externe systemen worden dan parallel (asynchroon) aangeroepen en de Workflow zorgt weer voor de coördinatie om de volgende activiteit uit te voeren. Dit komt uiteraard de performance ten goede.

Om de instellingen voor het mailen goed te zetten heeft de Mail-Activity twee event-handlers (één voor het verzenden en één erna). In de Workflow Designer kan je de handlers in je code-class genereren door te rechtsklikken en de optie 'Generate Handlers' te kiezen. In de handler voor het verzenden kan je de desbetreffende velden vullen. Zie de code in de download voor de details hoe dit werkt.

Vervolgens gebruiken we een GetData Activity om een synchrone callback te doen naar de Windows-applicatie. Om dit te simuleren heb ik in de Winforms-applicatie een methode geïmplementeerd die wordt aangeroepen door de Windows Workflow Foundation en at random het artikel goedkeurt of afkeurt. De class die deze

logica implementeert, moet de interface `IDataHandler` implementeren; zie codevoorbeeld 2. Er wordt een object verstuurd tussen de Windows Workflow Foundation en de runtime van een bepaald serializable type dat je zelf kunt implementeren; in mijn geval met slechts twee velden, het `reviewResult` en het onderwerp (`Subject`). Alleen het `reviewResult`-veld wordt gebruikt om het resultaat terug te geven. Doordat deze communicatie synchroon is, is het nog niet nodig een `correlation-field` op te geven, maar bij workflows die asynchroon en long running zijn komt dit, net als bij BizTalk Orchestrations, wel om de hoek kijken. Het `correlation-field` wordt gebruikt om de juiste instantie van de workflow terug te vinden en te activeren.

Zo zie je dat het eenvoudig is om synchrone communicatie op te zetten tussen de Winforms-applicatie (de host) en de Windows Workflow Foundation. Om de `GetData` Activity te laten werken heb je een `DataSourceService` nodig die effectief zorgdraagt voor de communicatie tussen de host en de workflow. In codevoorbeeld 3 zie je de code die draait na het drukken op de verstuurknop. Deze code start de Windows Workflow Foundation-runtime en initialiseert de `DataSourceService`. In een echte situatie zou je deze functionaliteit vanzelfsprekend implementeren door gebruik te maken van services en mogelijk gebruikmaken van Windows Communication Foundation. De `ListenActivity` en de `Listen Communication Services` gebruik je dan voor de communicatie. De `Data Services` gebruiken de `Listen Communication Services` onder de motorkap. Helaas kan ik in dit artikel niet dieper ingaan op de `Listen Communication Services`.

```
private void CodeWordCountHandler(object sender, EventArgs e)
{
    const string regexString = @"[\s\,\.\?!\'\""\(\)\;\:]+";
    string Text = this.Parameters["Text"].Value.ToString();
    Regex regexSplit = new Regex(regexString);
    amountOfWords = regexSplit.Split(Text).GetLength(0);
    MessageBox.Show("Aantal woorden: " + amountOfWords.ToString());
}
```

Codevoorbeeld 1.

Wordcount-algoritme

```
public class ReviewResultDataHandler : IDataHandler
{
    public object SelectData(object message, DataSourceInfo data)
    {
        // Het inkomende bericht is onze ReviewResult
        ReviewResult result = message as ReviewResult;

        // Met een kans van 80% 'reviewen' we het document
        Random random = new Random();
        int number = 0;
        lock (random)
        {
            number = random.Next(1, 100);
        }
        // We zetten de property met het resultaat.
        result.Result = number > 20;

        // We geven het resultaat terug aan de workflow.
        return result;
    }

    public void UpdateData(object message, DataSourceInfo data)
    {
        throw new NotImplementedException();
    }
}
```

Codevoorbeeld 2.

Goed- of afkeuren van het artikel

Afhankelijk van het resultaat eindigt de workflow of wordt het artikel doorgestuurd naar de redactie. De redactie doet in werkelijkheid een soortgelijke review als ik hierboven al heb beschreven. Nu stuurt deze activity alleen een mail naar de redactie, zodat ik nog wat ruimte heb om wat meer te schrijven over de overige activiteiten in Windows Workflow Foundation.

Nu heb ik twee approval-processen geïmplementeerd, waarbij mensen betrokken zijn. In veel workflow-processen is vooraf echter niet bekend hoeveel mensen moeten reviewen. Bovendien zal in veel gevallen pas tijdens het lopen van de workflow bekend zijn wie moeten reviewen, of mensen die vergeten te reageren. Niets is veranderlijker dan de mens, dus Microsoft heeft de ondersteuning hiervan een doel op zich gemaakt. Met de `Replicator Activity` is het mogelijk een Activity inclusief de onderliggende Activities-runtime te repliceren. Met de `Delay Activity` is het mogelijk om mensen te herinneren of alternatieve activiteiten uit te voeren, zoals zijn of haar baas verwittigen of het verzoek naar een collega sturen. Naast de al genoemde activiteiten zijn er ook nog activiteiten voor communicatie over webservices, transacties inclusief compensation, exception handling en (data) communicatie. Daarnaast is het doel van Microsoft dat, net als voor Presentatie-control voor Win- en Webforms het geval is, er third-party-controls uitgebracht gaan worden die je op kunt nemen in je applicatie. Microsoft doet dit vanzelfsprekend zelf ook met onder andere Office, maar ook met Dynamics (de paraplu waaronder de toekomstige versies van de Microsoft Business Solutions gaan vallen). De community site [www.windowsworkflow.net](http://www.windowsworkflow.net) wordt de [www.asp.net](http://www.asp.net) van Windows Workflow Foundation. Daar is al een aantal custom-activiteiten te vinden, zoals de mail Activity die ik heb gebruikt.

Indien langdurige transacties van toepassing zijn, dan ondersteunt Windows Workflow Foundation het zogenaamd 'dehydrateren' van Windows Workflow Foundation-instanties en kan de runtime deze op gezette momenten later weer tot leven wekken om verder te verwerken. Dit proces is vergelijkbaar met het dehydrateren van BizTalk Server Orchestrations. Standaard wordt State opgeslagen in SQL Server, maar ook hier is gedacht aan uitbreidbaarheid.

## Wanneer kan ik aan de slag?

Windows Workflow Foundation wordt uitgebracht als onderdeel van WinFX in midden 2006 en is geschikt om te draaien op Windows Vista, maar ook op Windows XP, Windows Server 2003

```
private void butSend_Click(object sender, EventArgs e)
{
    if (workflowRunTime == null)
    {
        workflowRunTime = new WorkflowRuntime();
        workflowRunTime.StartRuntime();
    }
    Assembly assembly = Assembly.Load("NetMagazineWorkflow");
    Type workflowType = assembly.GetType("NetMagazineWorkflow.Workflow1");

    Dictionary<string, object> parameters = new Dictionary<string, object>();
    parameters.Add("Title", txtTitle.Text);
    parameters.Add("SubTitle", txtSubTitle.Text);
    parameters.Add("Author", txtAuthor.Text);
    parameters.Add("Text", txtText.Text);

    DataSourceService service = new DataSourceService(workflowRunTime);
    workflowRunTime.AddService(service);

    workflowRunTime.StartWorkflow(workflowType,
        System.Guid.NewGuid(), parameters);
}
```

Codevoorbeeld 3.

Starten van de Windows Workflow Foundation-runtime en initialisatie van de `DataSourceService`

en de toekomstige Windows Longhorn Server-edities. Voor het bouwen van deze applicatie heb ik gebruik gemaakt van Windows Workflow Foundation die wordt gedistribueerd als onderdeel van de WinFx September CTP. Mijn advies is om nieuwe applicaties met workflow-functionaliteit te baseren op Windows Workflow Foundation. Samen met Windows Communication Foundation en Windows Presentation Foundation wordt het ons nog makkelijker gemaakt om robuuste bedrijfsapplicaties te implementeren. Helaas is WinFX vooralsnog nog in bètafase, maar mijn ervaring met het framework is dusdanig goed dat ik zeker niet afraad om al in de bètafase applicaties te gaan ontwikkelen. Het hangt helemaal van de timing (en deadlines) van je project af of het de moeite waard is gebruik te maken van de kracht van Workflow en WinFX.

**Dennis Mulder** is principal consultant bij **Avanade** ([www.avanade.com](http://www.avanade.com)), een samenwerkingsverband van Microsoft en Accenture. Voor vragen en opmerkingen is Dennis te bereiken op [dennism@avanade.com](mailto:dennism@avanade.com) of via zijn blog <http://www.dennismulder.net/blog>.

#### Referenties

<http://www.dennismulder.net/blog>

<http://www.avanade.com>

<http://msdn.microsoft.com/windowsvista/building/workflow/default.aspx>

<http://msdn.microsoft.com/library/en-us/dnlong/html/WWFIntro.asp>

<http://blogs.msdn.com/pandrew/>

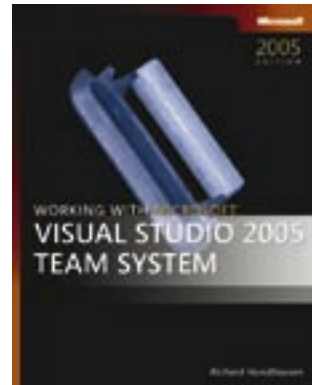
<http://msdn.microsoft.com/library/en-us/dnlong/html/WWFGetStart.asp>

<http://www.windowworkflow.net/>

<http://msdn.microsoft.com/workflow>

<http://msdn.microsoft.com/winfx/>

( advertentie Microsoft Press )



#### **Working with Microsoft® Visual Studio® 2005 Team System**

ISBN: 0-7356-2185-3

Auteur: Richard Hundhausen

Pagina: 336