

Het Sharepoint 2007 platform is zeer geschikt als ontwikkelplatform en als applicatiehosting-omgeving voor maatwerkapplicaties. Achmea heeft dit vroegtijdig onderkend en heeft het daarom als strategisch doelplatform geselecteerd. Een nadeel bij Sharepoint echter is de complexe(re) deployment, vergeleken met bijvoorbeeld standaard .NET-applicaties. Door toepassing van inherente Sharepoint-functionaliteit zelf is het mogelijk dat nadeel in een voordeel om te zetten. In dit artikel beschrijft de auteur hoe dit in een maatwerkapplicatieproject is bereikt.

Realiseren van maatwerk-webapplicaties

Sharepoint 2007 publishing-site deployment en provisioning

Het doel van het project is functioneel, namelijk het opleveren van gewenste functionaliteiten en conformeren aan de gestelde kwaliteitseisen (security, beschikbaarheid, robuustheid). Dit artikel beschrijft het technische hulpmiddel dat parallel daaraan is gerealiseerd. Dit hulpmiddel wordt gebruikt om gedurende de verschillende projectfasen het functionele projectresultaat te deployen door de OTAP-straat. Voor het onderwerp van dit artikel is de applicatiefunctie verder niet van invloed, en daarom niet beschreven.

Typische Sharepoint-deployment

Een Sharepoint-deployment kan getypeerd worden als handwerk met een diversiteit aan acties en is daarom foutgevoelig. Enkele voorbeelden van uit te voeren acties zijn de deployment van assemblies naar de global assembly cache (via gacutil) of naar de lokale virtual directory van de Sharepoint-webapplicatie (via xcopy). Andere voorbeelden zijn de toevoeging van webpart-files aan de Sharepoint webpartgallery, en de branding, zoals het deployen van custom masterpage en page lay-outs in de Sharepoint contentdatabase. De site aanpassen aan het runtime-gebruik van deze customization artefacts, xcopy van user controls, resources, en het aanpassen van de web.config voor SafeControls, trust level, en membership provider, zijn andere voorbeelden.

Dit handwerk is door middel van scripting op zich wel te automatiseren. Hiermee blijft het echter nog wel foutgevoelig. De belangrijkste redenen zijn dat de deployment-oplevering naast het script uit een fors aantal afzonderlijke bestanden/deliverables bestaat. Hierdoor is de deployment-unit niet eenduidig te identificeren. Een tweede reden is dat er handmatige activiteit op Sharepoint- en ASP.NET-vlak voor afconfigureren resteert. Voorbeelden zijn het zetten van de site masterpage, trust level in web.config enzovoort. Een derde reden ten slotte is dat deze werkwijze geen ondersteuning geeft voor webfarms. De hosting-afdeling moet zelf de deployment-uitvoering identiek herhalen over iedere afzonderlijke server in de farm.

Gewenste deployment-situatie

De gewenste deployment-situatie is als volgt onderkend. Er is sprake van een *volledig geautomatiseerde* deployment van de Sharepoint-maatwerkapplicatie. Daarmee bedoelen we inderdaad dat deze volledig geautomatiseerd is. Er zijn geen handmatige acties vooraf of achteraf op de hosting-omgeving. Er wordt in één keer gedeployed, zowel voor single server- als webfarm-omgeving. De oplevering bestaat uit één op zichzelf staand deployment-package. De deployment is omkeerbaar, wat wil zeggen dat de initiële situatie op de hosting-omgeving vol-

William van Strien

is .NET softwarearchitect bij Ordina ICT – Microsoft Solutions. Het resultaat in dit artikel is mede ontstaan vanuit zijn inzet bij Achmea Group IT Services.

William is te bereiken via william.van.strien@ordina.nl of w.van.strien@interpolis.nl

ledig kan worden gerestored. De deployment is herhaalbaar door de gehele OTAP-straat. Op alle onderdelen wordt hetzelfde deployment-package toegepast voor de inrichting. Er is sprake van een automatisch inrichting van de Sharepoint-site wat betreft customization/branding en content. Tot slot betekent dit dat de deployment-package zelf geautomatiseerd kan samenstellen/assembleren, en dit onder andere kan worden uitgevoerd in de context van de nightly build.

Sharepoint Solution framework en features

Het Sharepoint-platform biedt zelf ondersteuning om de gewenste deployment-situatie te realiseren. Dit bestaat uit twee onderdelen: het Sharepoint Solution-framework en de Sharepoint Features. Een gedetailleerde beschrijving en uitleg van deze Sharepoint-onderdelen is onder andere te vinden op MSDN. Voor het begrip van dit artikel is het voldoende om ze op hoofdlijnen te beschrijven. Door het Solution-framework kun je volledig geautomatiseerd en gecontroleerd de verschillende applicatieonderdelen deployen naar de doel Sharepoint hosting-omgeving. Het framework zorgt er daarbij zelf voor dat dit herhaald wordt over alle servers in de webfarm. Voorbeelden van applicatieonderdelen zijn hierbij de custom masterpage, eigen pagelayout(s), en eventueel contenttypes; webparts; resources, zoals documenten, plaatjes, en resourcefiles; user controls + codebehind; applicatiepages en codebehind, en Sharepoint Features.

Sharepoint-features zijn in feite de standaard manier om een Sharepoint runtime-omgeving aan te vullen met extra capabilities. Het Sharepoint-platform, WSS en MOSS, past dit mechanisme zelf toe om de basisinrichting verder aan te vullen. Voorbeelden hiervan zijn de standaard search-functionaliteit en teamcollaboratie. Features kun je activeren en ook weer deactiveren om de toegevoegde functionaliteit te verwijderen of te disablen. Dit kan op drie niveaus: server-breed, site-collection en site-level (individuele webapplicatie). Ook custom functionaliteit kun je als feature toevoegen aan de Sharepoint-doelomgeving en, indien van toepassing, later weer deactiveren. Zeker gedurende de realisatie en de testfasen voor de O+T-omgevingen is dit een gewilde feature.

Inhoudelijke beschrijving

In het project is de deployment en provisioning via twee hoofd-niveaus gerealiseerd. *Niveau 1:* de deployment van de applicatieonderdelen naar de Sharepoint-omgeving(en) vindt volledig automatisch en autonoom plaats door toepassing van het Solution framework. Door sturing via de manifest-file zorgt het Solution-framework dat iedere

individuele deliverable op de juiste doellocatie belandt. Dat zijn de Sharepoint contentdatabase, GAC, 12hive of de lokale bin van de Sharepoint-webapplicatie.

```

<?xml version="1.0" encoding="utf-8" ?>
<Solution xmlns="http://schemas.microsoft.com/sharepoint/"
  SolutionID="5DA7E1B-C405-4f69-A354-5B72010BE19"
  DeploymentServerType="WebFrontEnd">
  <Assembly>
    <Assembly DeploymentTarget="GlobalAssemblyCache" Location="EALFProvisioningFeature\Assemblies\Interpolis.GITS.EALF.SharePointDefinition.dll"/>
    <Assembly DeploymentTarget="GlobalAssemblyCache" Location="EALFWebPartsFeature\Assemblies\Interpolis.GITS.EALF.Common.dll"/>
    <Assembly DeploymentTarget="GlobalAssemblyCache" Location="EALFWebPartsFeature\Assemblies\Interpolis.GITS.EALF.Modules.dll"/>
    <Assembly DeploymentTarget="GlobalAssemblyCache" Location="EALFWebPartsFeature\Assemblies\Interpolis.GITS.EALF.Presentation.dll"/>
    <Assembly DeploymentTarget="GlobalAssemblyCache" Location="EALFWebPartsFeature\Assemblies\Interpolis.GITS.EALF.Views.dll"/>
    <SafeControlID [redacted]
      <SafeControlID [redacted]
    </Assembly>
    <Assembly DeploymentTarget="GlobalAssemblyCache" Location="EALFWebPartsFeature\Assemblies\ReturnOfHeartPart.dll"/>
    <Assembly DeploymentTarget="WebApplication" Location="Interpolis.GITS.EALF.HI.dll"/>
  </Assembly>
  </Assemblies>
  <TemplateFiles>
    <TemplateFile Location="CONTROLTEMPLATES\EALFUserLogin.aspx" />
    <TemplateFile Location="CONTROLTEMPLATES\SubjectSubscribe.aspx" />
    <TemplateFile Location="CONTROLTEMPLATES\NewsCommentInput.aspx" />
  </TemplateFiles>
  <RootFiles>
    <RootFile Location="TEMPLATE\LAYOUTS\1033\GITSscore.js" />
    <RootFile Location="TEMPLATE\LAYOUTS\EALFAppSettings.aspx" />
  </RootFiles>
  <FeatureManifests>
    <FeatureManifest Location="EALFProvisioningFeature\Feature.xml" />
    <FeatureManifest Location="EALFWebPartsFeature\Feature.xml" />
  </FeatureManifests>
</Solution>

```

Figuur 1: Sharepoint Solution Manifest-file

Niveau 2: onderdeel van de Solution-deployment zijn twee features. De eerste zorgt bij activatie voor de toevoeging van geïdentificeerde maatwerk-webparts aan de webpart-gallery van de doel Sharepoint-site. De tweede feature behandelt bij activatie de provisioning van de doel Sharepoint-site: customization/branding, aanmaken content, doorvoeren wijzigingen in de web.config. Het aanmaken van content bestaat in de huidige versie uit de volgende onderdelen:

- Aanmaken van lijsten, inclusief columns en fields. Ook de generatie van look-up fields naar een andere lijst is hierbij ondersteund.
- Als specialisatie: document- en picture-libraries, en discussieboards.
- Content-pages
 - Gebaseerd op gespecificeerde pagelayout, en dus content type. Dit kan een standaard pagelayout zijn; in het project betreft het vanwege branding in alle gevallen de eigen pagelayouts.
 - Automatisch opnemen van webparts, in de aangeduide webpart zone. De juiste webpart wordt zelf runtime bepaald uit de webpart gallery, en dan via reflection geïnstantieerd. Dit is de reden om te splitsen in twee afzonderlijke features.
 - Leggen van webpartconnecties tussen twee webparts op dezelfde page: provider en consumer.
 - Wel/niet opnemen van een page in de Sharepoint-navigatiemenu's.

De provisioning-uitvoering zelf is generiek opgezet, en daarmee herbruikbaar voor andere Sharepoint-applicaties en projecten. De applicatiespecifieke provisioning is gespecificeerd in een XML-file.

```

    <Property Naam="Location" Type="string" Value="" />
  </Properties>
</Webpart>
</Webparts>
</Page>
<Page Naam="News.aspx" Template="EALPWelcomeSplash.aspx" Description="Nieuws" IncludeInGlobalNavigation="true" />
<Page Naam="NewsItem.aspx" Template="EALPLinkPopupWindow.aspx" Description="EALP nieuwsbericht" IncludeInGlobalNavigation="true" />
<Webparts>
  <Webpart ID="MainNews" Naam="NewsItemWebPart web part" WebPartZone="TopCenter" Position="1" Type="NewsItemWebPart" />
  <Webpart Naam="ReactieOpNieuws" WebPartZone="TopCenter" Position="2" Type="SmartPartv3_1_2" Chronology="1" />
  <Properties>
    <Property Naam="UserControl" Type="string" Value="~/_controltemplates/NewsCommentInput.aspx" />
    <Property Naam="SerializedProperties" Type="string" Value="" />
  </Properties>
</Webpart>
  <Webpart ID="RelatedNews" Naam="RelatedNews web part" WebPartZone="TopRight" Position="1" Type="RelatedNewsWebPart" />
</Webparts>
<WebpartConnections>
  <WebpartConnection Provider="MainNews" ProviderConnectionPoint="AddressProvider_RelatedNews" Consumer="ReactieOpNieuws" />
</WebpartConnections>
</Page>
<Page Naam="Courses.aspx" Template="EALPWelcomeSplash.aspx" Description="Opleidingen" IncludeInGlobalNavigation="true" />
<Webparts>
  <Webpart Naam="Opleidingheader web part" WebPartZone="TopCenter" Position="1" Type="ContentEditorWebPart" />
  <Properties>
    <Property Naam="Image" Type="String" Value="" />
  </Properties>
</Webpart>
</Webparts>

```

Figuur 2: de hoofdelementen van de site-provisioning-specificatie

Bij het opzetten van de siteprovisioning-engine is er bewust aandacht aan geschonken om bestaande content te kunnen behouden. De waarde of zelfs noodzaak hiervan voor de productieomgeving zal duidelijk zijn. De eerlijkheid echter gebiedt te melden dat het in eerste instantie uit eigenbelang is geboren. Het is namelijk bijzonder vervelend om in jouw lokale Sharepoint-ontwikkelingsomgeving voortdurend de opgebouwde testdata kwijt te raken bij een redeployment. Want het projectteam past zelf ook voortdurend de deployment en provisioning toe in de individuele eigen lokale ontwikkelwerkplek.

Ervaringen

Buiten wat kinderziekten bij het opzetten, uitbouwen en toepassen van het beschreven mechanisme zijn de ervaringen louter positief. De hosting-afdeling waardeert het zeer dat er hiermee een gestructureerde en volledig geautomatiseerde deployment en inrichting van de Sharepoint-applicatie is, eentje die ze ongewijzigd door de gehele OTAP kunnen trekken. Ook voor het ontwikkelteam is het bijzonder prettig om op dezelfde wijze de eigen omgeving in te richten, en dit te herhalen wanneer dat nodig is. Nieuwe projectleden hebben hiermee ook met één actie de volledige applicatie lokaal ingericht en beschikbaar.

```

<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" />
  <Pages>...
  <DocLibraries>...
  <PictureLibraries>...
  <Lists>...
  <DiscussionBoards>...
  <WebSettings Title="EALP" Description="EALP" />
  <WebConfigSettings>...
  <ApplicationSettings>...
  <CleanupSettings>...
</Elements>

```

Figuur 3: een detailblik op de site-provisioning-specificatie

Nog andere technische functionaliteiten

Naast het beschreven technische raamwerk voor Sharepoint applicatie-site-deployment en provisioning zijn er ook nog wat andere technische zaken gerealiseerd om de gevraagde applicatiefunctionaliteit te maken. Er is sprake van een custom membership provider, met user-administratie in een SharePoint-lijst (met dank aan CodePlex). Sharepoint-properties kunnen worden toegepast voor het configureren van centrale applicatiesettings. Webparts ondersteunen uiteraard zelf ook de instelling via properties, maar de reikwijdte hiervan is beperkt tot de scope van webpart-instance op één content-page. Er is een custom roleprovider die de autorisatie tot bepaalde site-onderdelen dynamisch bepaald op basis van de toestand van een geauthenticeerde gebruiker; deze wordt beheerd in een externe gebruikersadministratie. Het reikt echter te ver om deze onderdelen hier ook nader te beschrijven.

Herhaalbare en gecontroleerde oplevering

Het Sharepoint 2007 platform blijkt inderdaad zeer geschikt voor realiseren van maatwerk webapplicaties. Het platform komt out-of-the-box met een rijkgevolde doos vol met technische en functionele bouwstenen. Hiermee kan de applicatieontwikkelaar snel typisch gevraagde functionaliteiten assembleren. Klantsituatiespecifieke zaken kun je dan verder realiseren via maatwerk .NET-onderdelen. Het deployen en inrichten van de Sharepoint-applicatie op de hosting-omgeving stelt vanuit de Sharepoint-architectuur en opzet specifieke eisen, en is breder en complexer dan bij een standaard ASP.NET-webapplicatie. Het Sharepoint-platform levert ook hiervoor ondersteuning om dit proces volledig geautomatiseerd uit te voeren. In dit artikel is beschreven dat op basis hiervan een generiek raamwerk is gerealiseerd waarmee herhaalbaar en gecontroleerd Sharepoint-applicaties in de OTAP-straat worden opgeleverd en ingericht. «