

Met behulp van Content Management Server 2002 (CMS) kun je eenvoudig te onderhouden websites bouwen en beheren. De business driver van CMS is het gebruiksvriendelijk beheer van een website, in het bijzonder voor mensen zonder kennis van programmeren of HTML. CMS maakt het mogelijk dat bijvoorbeeld secretaresses nieuwe informatie online kunnen zetten. Voordat de secretaresses aan de slag gaan met de inhoud van een website zullen wij – software ontwikkelaars – voorbereidingen moeten treffen. Een groot deel van deze voorbereiding zit in het bouwen van templates. Templates brengen een scheiding aan tussen content en de presentatielaag van een website.



thema

Programming Content Management Server 2002

Krachtige applicatieserver door integratie met Visual Studio

Vaak bestaan features van een website uit ontsluiting van specifieke databases, maatwerk. Dit verandert niet door CMS. In veel gevallen is het verstandig de mogelijkheden van CMS templates te combineren met maatwerk. Er zijn ontwikkelaars die weerstand hebben tegen het gebruik van een CMS omgeving. Sommigen bouwen het liefst zelf een content management omgeving om alles in de hand te houden. De vergaande integratie van CMS met Visual Studio .NET zorgt ervoor dat de programmeur volledige controle kan blijven uitoefenen vanuit één vertrouwde omgeving. Figuur 1 geeft een overzicht van de CMS architectuur.

Een aantal bouwstenen zal in dit artikel onder de loep worden genomen. Het kloppende hart van CMS is een ISAPI filter dat HTML bedient aan de verschillende gebruikers. Alle data die CMS gebruikt, worden opgeslagen in een SQL Server database. De Content Connector voor een verbinding met Microsoft Commerce Server en Site Deployment vallen buiten de scope van dit artikel. De Authoring Connector zorgt ervoor dat content managers eenvoudig vanuit Word content kunnen toevoegen aan CMS. Voordat we dieper ingaan op het programmeren van ASPX templates is het belangrijk stil te staan bij de overige bouwstenen van de Content Management Server architectuur.

WEB AUTHOR CONTROL De belangrijkste feature van CMS is gebruiksvriendelijke web-based content

management. Web authors kunnen middels een browser content managen door gebruik te maken van de mogelijkheden van de zogenaamde web author control die onderdeel uitmaakt van de HTML pagina. Via de web author control kunnen nieuwe pagina's worden toegevoegd aan de website of bestaande worden gewijzigd. Andere mogelijkheden van de web author control zijn te zien in figuur 2.

TEMPLATE DEVELOPMENT Web authors kunnen al hun taken eenvoudig verrichten met behulp van een browser. Programmeurs van een website hebben de beschikking over Visual Studio .NET. Het belangrijkste middel voor programmeurs om een website te ontwikkelen zijn templates (ASP.NET pagina's). Content wordt via zogenaamde placeholders in de templates geïnjecteerd door CMS. De web author control wordt geplaatst op de template.

SITE MANAGER Om CMS te configureren wordt gebruik gemaakt van de site manager applicatie. Dit is een administratieve omgeving waar de navigatiestructuur, gebruikers authenticatie, workflow en website resources kunnen worden onderhouden. De navigatiestructuur van één of meerdere website(s) wordt weerspiegeld door zogenaamde channels. Een channel is vergelijkbaar met een folder, het is een verzamelbak voor - op templates gebaseerde - webpagina's, opgenomen in een boomstructuur. Deze structuur kan met

behulp van de Site Manager op vergelijkbare wijze worden onderhouden als folders middels Windows Explorer. Channels kunnen programmatisch worden benaderd, op deze wijze is het mogelijk dynamische website menu's te creëren.

Het authenticatie model kan worden ingericht door gebruikers toe te voegen aan content management rollen, waaronder template designers, editors, authors, moderators en subscribers. Met behulp van het authenticatie model kan nauwkeurig worden bepaald welke gebruikers specifieke operaties mogen uitvoeren op de website. Er kunnen meerdere groepen worden geconfigureerd binnen een bepaalde rol. Daarmee wordt het mogelijk een bepaalde groep via gebruikersrechten gericht toegang te geven. Zo kan bijvoorbeeld een groep website editors rechten krijgen op een specifiek gedeelte van de navigatiestructuur (channels). Hetzelfde geldt voor resources en channels. Authenticatie is onderdeel van de *Microsoft.ContentManagement.Web.Security namespace*.

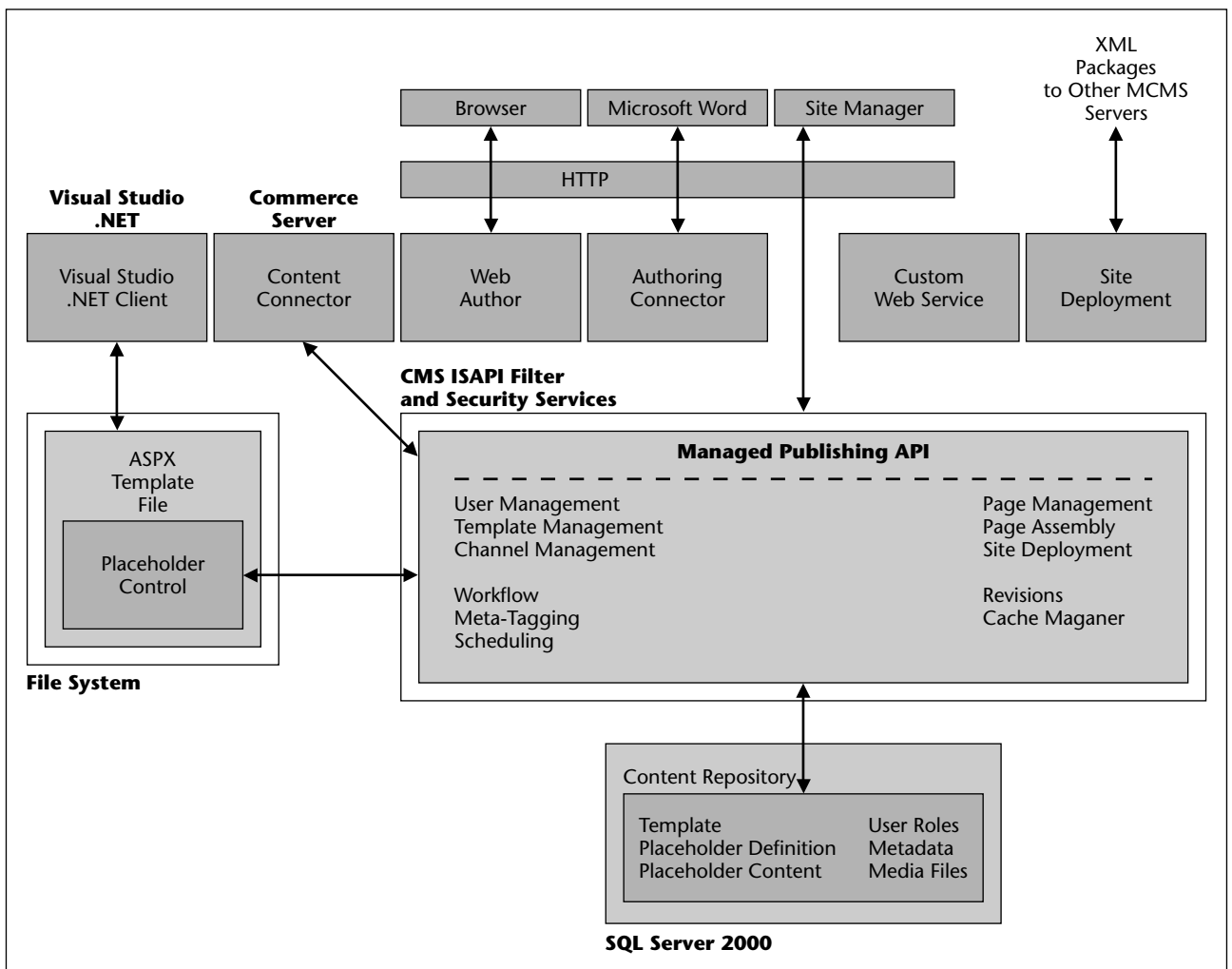
ROLLEN Het beheer van de content van een website wordt over het algemeen door een groep van personen

uitgevoerd. Om ervoor te zorgen dat deze personen op een effectieve manier met elkaar samenwerken zijn er in CMS workflow features geïmplementeerd. De eerder genoemde rollen werken op deze wijze samen:

- Stap 1 – Creatie template (template designer).
- Stap 2 – Toevoegen content (author).
- Stap 3 – Wijzigen content (editor).
- Stap 4 – Goedkeuring en publicatie (moderator).
- Stap 5 – Bekijken content (subscriber), bezoeker website.

Het workflow proces kan worden gestroomlijnd door gebruikmaking van e-mail notificatie. Uiteraard is het mogelijk de workflow te configureren opdat authors pagina's direct zelf kunnen publiceren zonder tussenkomst van een editor of moderator. Ook is het mogelijk het workflow proces vergaand aan te passen met behulp van de *Microsoft.ContentManagement.Publishing.Events namespace*. In de content management markt wordt vaak zwaar getild aan workflow features, in de praktijk valt de toepassing van workflow vaak tegen.

De Site Manager ondersteunt het centraal beheer van resources, bijvoorbeeld documenten die op meer plekken in de site terugkomen. Nieuwe versies van deze



FIGUUR 1. Overzicht van de architectuur van een content management systeem



FIGUUR 2. De web author control kan eenvoudig worden uitgebreid of aangepast.

resources hoeven slechts op één plek te worden gewijzigd. Resources management heeft met name betrekking op informatie die regelmatig wijzigt. Vaste onderdelen van de website kunnen beter hard gecodeerd worden opgenomen in de templates omdat resources management consequenties heeft voor de performance overhead.

VISUAL STUDIO .NET Bij de installatie van CMS 2002 worden er een aantal modules toegevoegd aan Visual Studio .NET. De eerste wijziging aan de CMS interface is de toevoeging van een aantal nieuwe projecttypes (zie figuur 3).

Het is mogelijk om een CMS webservice te maken die content ontsluit als een web services. Het aanmaken van MCMS Web Application is vergelijkbaar met een gewone ASP.NET web applicatie met die toevoeging dat een aantal zaken wordt toegevoegd. Bestaande ASP.NET web applicaties kunnen eenvoudig worden omgetoverd tot MCMS applicaties.



FIGUUR 3. Toevoeging van een aantal nieuwe projecttypes aan de CMS interface

Toevoegingen:

- 1 Referenties naar relevante Microsoft.ContentManagement namespaces.
- 2 Aanpassingen aan Web.Config.
- 3 Toevoeging standaard Web Author control.
- 4 Toevoeging van MCMS aan toolbox en placeholder controls (zie figuur 5).
- 5 Toevoeging MCMS template explorer (zie figuur 4).

PROGRAMMING INTERFACE Software ontwikkelaars kunnen gebruik maken van de rijke mogelijkheden van de Microsoft.ContentManagement namespace. De namespace is complementair aan het .NET framework, met name System.Web.

De Microsoft.ContentManagement namespace is een op .NET gebaseerde API die deels een wrapper is om bestaande CMS 2001 COM interfaces en deels native in .NET geschreven. De bestaande COM interfaces zijn in verband met backward compatibility nog steeds bruikbaar in CMS 2002. Hetzelfde geldt voor ASP dat naast ASP.NET gebruikt kan worden. De belangrijkste interfaces van CMS programmeurs liggen opgeslagen in de Microsoft.ContentManagement.Publishing namespace: Context, Channels, Postings, Templates en Placeholders.

CONTEXT In den beginne was er de Context; dit is het uitgangspunt. Vanuit de 'current' context kan de programmeur toegang krijgen tot de belangrijke CMS objecten. Er zijn twee soorten context: de CmsApplicationContext voor 'fat client' applicaties en de CmsHttpContext voor web applicaties. De context is static en hoeft dus niet te worden geïnstantieerd:

```
Dim cmsContext As CmsHttpContext =
    CmsHttpContext.Current
```

CHANNELS, POSTINGS EN PLACEHOLDERS De navigatiestructuur van CMS komt tot uitdrukking in Channels en Postings. Als je dit vergelijkt met Windows Explorer, dan zijn Channels directory's en Postings files. De 'current' context weet op welke plek in de navigatiestructuur een bepaalde gebruiker zich bevindt. Stel dat een gebruiker zich bevindt in de nieuwschannel van een website, gegeven dat de nieuwspagina anders is dan een opsomming van een aantal nieuws postings binnen de



FIGUUR 4. MCMS Template Explorer

nieuws channel. De volgende code toont de inhoud van de nieuws postings:

```
Dim newsPostings As PostingCollection
Dim newsPosting As Posting
Dim newsContent As Placeholder

newsPostings = cmsContext.Channel.Postings()
For Each newsPosting In newsPostings
    For Each newsContent In newsPosting.
        Placeholders
        Response.Write(newsContent.Data-
            source.RawContent)
    Next
Next
```

De inhoud van een Posting ligt opgeslagen in zogenaamde Placeholders. Eén Posting kan meerdere Placeholders bevatten. Dit is afhankelijk van de Template die ten grondslag ligt aan de Posting. De Channel en Posting interfaces zijn iteratief, met andere woorden: een Channel kan meerdere Postings bevatten, ook meerdere Channels. Deze set van onderliggende Channels of Postings zijn opgeslagen als een Collection en derhalve 'foreachable'. Er zijn uiteraard vele variaties denkbaar op bovenstaand voorbeeld, denk bijvoorbeeld aan de dynamische opbouw van een uitklapbaar DHTML menu geba-

biedt de programmeur een aantal bouwstenen die de content management features representeren. De belangrijkste bouwsteen zijn de PlaceholderControls, voor bijvoorbeeld HTML, plaatjes, attachments en XML. Web authors die een posting benaderen via een website kunnen content management taken verrichten door gebruik te maken van de mogelijkheden van het Web Author control. Daarvoor dient deze standaard CMS control te worden opgenomen op de template.

CUSTOM PROPERTIES Placeholders worden gebruikt om data (content) entry mogelijkheden toe te voegen aan templates. Custom properties kunnen worden gebruikt om meta-data toe te voegen aan postings en channels. Programmeurs kunnen deze meta-data uitlezen en verwerken. Deze feature biedt talrijke mogelijkheden. Ze kan bijvoorbeeld gebruikt worden om web authors de mogelijkheid te geven eenvoudig HTML meta-data toe te voegen aan postings. Dit kan er voor zorgen dat de CMS website hoger uitkomt bij zoekmachines.

PLACEHOLDERS Behalve de genoemde placeholders bestaat er ook de mogelijkheid om zelf placeholders te ontwikkelen. Net als het gebruik van custom properties bieden zelfgemaakte placeholders ruimte voor creatieve oplossingen. HTML, image en attachments placeholders voegen content entry mogelijkheden toe op gelijknamige wijze. Custom placeholders bieden de mogelijkheid om de content managers bepaalde zaken te laten selecteren uit bijvoorbeeld dropdown boxen in de authoring modus van een posting. Deze selectie kan bijvoorbeeld door een programmeur worden gebruikt om bepaalde zaken in de presentatie modus van een posting wel of niet te laten zien. Authoring modus is de view op de CMS website door de ogen van content managers, terwijl standaard website bezoekers in presentatie modus werken. Het is ook mogelijk meta-data op deze manier te laten toevoegen door content managers in plaats van via de custom properties. Het voordeel van placeholders ten opzichte van custom properties is dat zij zijn opgenomen op dezelfde pagina waar de web author content beheert en niet in een daaraan gekoppelde property window. Een ander voordeel is de mogelijkheid om gebruik te maken van ASP.NET server controls zoals dropdown boxen.

Het is mogelijk om zelf Placeholder controls te maken door van de BasePlaceholderControl class te overerven (Microsoft.ContentManagement.WebControls). Voordat je een custom placeholder kunt compileren zul je vijf methodes moeten implementeren:

- CreateAuthoringChildControls, CreatePresentationChildControls; methoden voor creatie van de controls die opgenomen worden op de custom placeholder container.

Het kloppende hart van CMS is een ISAPI filter dat HTML bedient aan de verschillende gebruikers

seerd op Channels en Collections. Hergebruik van zo'n dynamisch menu in meerdere templates kan worden gerealiseerd met ASP.NET user controls.

TEMPLATE De file waarop een programmeur zijn kunsten kan loslaten is een template. Een template is evenals de Postings, Channels en Placeholders opgenomen in de CMS database. De programmeur werkt echter met een ASPX file die gekoppeld is aan deze database template. Deze koppeling wordt tot stand gebracht met behulp van de eerder genoemde Template Explorer. Het is aan de programmeur om invulling te geven aan de ASPX file. CMS



FIGUUR 5. MCMS Toolbox en placeholders

- LoadPlaceholderContentForAuthoring, LoadPlaceholderContentForPresentation; methoden voor het laden van content in de controls.
- SavePlaceholderContent; een methode om de content op te slaan in de CMS database.

De dubbele methodes voor creatie en het laden van content hebben betrekking op enerzijds authoring modus voor content managers en anderzijds presentatie modus voor reguliere site bezoekers. Als je een custom placeholder control hebt gemaakt kan deze worden opgenomen in de VS.NET Toolbox. De control kan in CMS worden opgenomen door deze van de Toolbox te slepen op een template ASPX pagina.

TEXTBOX Hier volgt een simpel voorbeeld waar een textbox web control wordt toegevoegd aan een custom placeholder in authoring modus en een label control in presentatie modus. De inhoud die de content manager opneemt in de textbox zal aan site bezoekers worden getoond.

```
Inherits Microsoft.ContentManagement.Web-
Controls.BasePlaceholderControl

Private authoringText As System.Web.UI.Web-
Controls.TextBox
Private presentationText As System.Web.UI.-
WebControls.Label

Protected Overrides Sub CreateAuthoringChild-
Controls(ByVal authoringContainer As
Microsoft.ContentManagement.WebControls.BaseMo-
deContainer)
    authoringText = New System.Web.UI.Web-
Controls.TextBox()
    authoringText.ID = "AuthoringControl"
    authoringText.TextMode = System.Web.
UI.WebControls.TextBoxMode.SingleLine
    authoringText.Height = New Sys-
tem.Web.UI.WebControls.Unit(200)
    authoringText.Width = New Sys-
tem.Web.UI.WebControls.Unit(200)
    authoringContainer.Controls.Add(autho-
ringText)
End Sub

Protected Overrides Sub CreatePresentation-
ChildControls(ByVal presentationContainer As
Microsoft.ContentManagement.WebControls.BaseMo-
deContainer)
    presentationText = New System.Web.
UI.WebControls.Label()
    presentationText.ID =
"PresentationControl"
    presentationContainer.Con-
trols.Add(presentationText)
End Sub
```

Vervolgens kunnen deze controls worden gevuld met content uit de CMS repository. Tevens is het mogelijk wijzigingen op te slaan in de CMS database.

```
Protected Overrides Sub LoadPlaceholder-
ContentForAuthoring(ByVal e As Microsoft.Con-
tentManagement.WebControls.PlaceholderControlE-
ventArgs)
    authoringText.Text = BoundPlace-
holder.Datasource.RawContent
End Sub

Protected Overrides Sub LoadPlaceholder-
ContentForPresentation(ByVal e As
Microsoft.ContentManagement.WebControls.Place-
holderControlEvents)
    presentationText.Text = BoundPlace-
holder.Datasource.RawContent
End Sub

Protected Overrides Sub SavePlaceholder-
Content(ByVal e As
Microsoft.ContentManagement.WebControls.Place-
holderControlSaveEventArgs)
    BoundPlaceholder.Datasource.RawContent
    = authoringText.Text
End Sub
```

Voordat de custom placeholder control op de template zijn krachten kan tonen zal de PlaceholderToBind property van de custom placeholder gekoppeld moeten worden aan een PlaceholderDefinition zoals opgenomen in de template via de template explorer. Deze koppeling is nodig om de verbinding met de CMS repository te bewerkstelligen. Hierin wijkt de custom placeholder niet af van de overige placeholders.

CONCLUSIE Content Management Server 2002 is door de vergaande integratie met Visual Studio .NET een zeer krachtige applicatieserver geworden. De CMS objecten vormen een welkome aanvulling op het .NET framework. De toetredingsdrempel voor programmeurs om gebruik te maken van CMS 2002 in combinatie met Visual Studio .NET is laag en biedt de mogelijkheid om niet alleen een scheiding aan te brengen tussen vormgeving en applicatie logica (*code behind*) maar ook tussen de vormgeving en content. Ik stel voor dit *content behind* te noemen. De meeste websites zijn gebaat bij content behind. Als deze baten opwegen tegen de licentiekosten van CMS dan is een keuze snel gemaakt.

Edwin Jongsmā