

Gestandaardiseerde werkwijze wordt afgedwongen

ENTERPRISE MASHUP TOOLS

De aandacht voor mashups is aanzienlijk toegenomen. Gartner geeft in zijn hype cycle aan dat de technologie zich midden in de hype bevindt. Maar wat is een mashup nu eigenlijk? Wat kun je er technisch mee? Wat hebben bedrijven eraan? Is een mashup slechts een hype of biedt deze technologie meer?

Door Edwin van Asch en Michiel Kemperman

Op internet zijn talloze voorbeelden te vinden van zogenaamde mashups voor consumenten. Voorbeelden hiervan zijn iGoogle, Netvibes en Pageflakes. Dit zijn zogeheten online mashups waarbij gebruikers de inhoud zelf kunnen bepalen. Bij iGoogle is het mogelijk om uit een aantal componenten te kiezen die we in een mashup willen hebben. Deze componenten worden ook wel *widgets* of *gadgets* genoemd. Iedere widget representeert functionaliteit of data van een ander systeem. Denk hierbij aan nieuws van een website (in de vorm van een RSS-feed), een kalenderfunctie, een to-do-lijstje, de weersverwachtingen van een bepaalde locatie etcetera. Dit is eigenlijk de meest eenvoudige vorm van een mashup. De mashup-elementen worden naast elkaar getoond en er vindt geen interactie plaats tussen de verschillende componenten.

Mashups voor consumenten

In een meer geavanceerde mashup-omgeving worden de verschillende componenten gemixt. Verschillende frameworks bieden de mogelijkheid om in een webpagina eenvoudig verschillende externe bronnen op te nemen en te mengen. Een eenvoudig voorbeeld is een mashup van meerdere RSS-feeds over hetzelfde onderwerp die worden

gecombineerd, gefilterd en gesorteerd en vervolgens worden weergegeven in één pagina. Een ander voorbeeld is een Google-map gecombineerd met foto's van de fotowebsite Flickr. De foto's worden geprojecteerd op een kaart aan de hand van de plek waar de foto is genomen.

De tools die in deze categorie vallen zijn bijvoorbeeld Microsoft Popfly, Yahoo Pipes en de Google Web Toolkit. Met deze tools zijn snel nieuwe applicaties te bouwen die zijn opgebouwd uit externe componenten. Hierbij wordt gebruik gemaakt van de gepubliceerde interfaces van deze componenten (de zogenaamde API's) die op open standaarden zijn gebaseerd zoals RSS, REST, SOAP etcetera. De tot nu toe genoemde voorbeelden zijn allemaal opgebouwd uit op de consument georiënteerde technieken. In een bedrijfsomgeving zijn deze over het algemeen minder bruikbaar, omdat hier andere eisen worden gesteld aan mashup-oplossingen.

Geavanceerde enterprise mashups

Lang niet alle bronnen die wenselijk zijn, kunnen direct worden opgenomen in een mashup. Reden hiervoor is dat ze dikwijls niet zijn voorzien van een geschikte API die voldoet aan open standaarden. Bij mashups voor consumenten gaat het meestal om het bij elkaar brengen van verschillende op internet gebaseerde bronnen die zijn voorzien van een API die voor iedereen toegankelijk is. Het is hierbij essentieel dat de

mashup wordt uitgevoerd op een grafisch aantrekkelijke manier. Bij enterprise mashups is dit echter van veel minder belang.

In een enterprise-omgeving is de mashup gericht op het integreren van een combinatie van interne legacy-systemen met externe (internet- of intranet-) bronnen. De technieken die in de mashup-omgeving voor consumenten voorhanden zijn, zijn echter niet toereikend om de bestaande (legacy) systemen onder te brengen in een mashup.

Zelfs als deze technieken wel toereikend zouden zijn, is het nog steeds niet mogelijk om mashups op grote schaal te gebruiken in een enterprise-omgeving. De reden hiervoor is dat de mashups die met de huidige consumenten frameworks worden gerealiseerd volledig voorbijgaan aan de traditionele integratieproblematiek waarmee bedrijfsomgevingen worden geconfronteerd. Het maken van een enterprise mashup is immers niets anders dan het integreren van de gebruikersinterfaces van applicaties of onderdelen daarvan. De mashup-omgeving voor een enterprise mashup moet de snelheid en wendbaarheid van de mashup-omgevingen voor consumenten combineren met de robuustheid, kwaliteit en functionaliteit die nodig is voor het integreren van alle mogelijke legacy-systemen.

SOA versus enterprise mashup

Het integreren van systemen heeft zich door het gebruik van veel specifieke interfaces, messaging middleware en EAI-platforms in de loop der jaren ontwikkeld tot de huidige stand van zaken: service oriëntatie. Service oriëntatie biedt daarentegen geen oplossing voor het realiseren van een geïntegreerde user interface zoals we die bij een mashup tegenkomen. Het gebruik van service oriëntatie is bij de user interface-laag niet meer toepasbaar. De user interfaces worden in het huidige SOA-tijdperk nog vaak op een traditionele manier gecodeerd in de vorm van bijvoorbeeld J2EE- of .NET-applicaties. Maar waarom zijn de SOA-principes en -componenten niet geschikt voor de gebruikers-interface? Services die in een reguliere SOA-implementatie worden gebruikt, hebben vooral betrekking op (business) functionaliteit en data. Deze services beschikken echter niet over een user interface. De vraag is dus of het mogelijk is om een service georiënteerde aanpak te volgen die dient als basis voor het realiseren van een gebruikersinterface. De enterprise mashup biedt hiervoor een uitstekend platform.

Wat houdt het toepassen van de principes van service oriëntatie bij het realiseren van een enterprise mashup nu precies in? Een enterprise mashup wordt volgens SOA-principes opgebouwd uit services. Deze zogenaamde user interface services zijn goed vergelijkbaar met de bekende SOA services. In beide gevallen geldt dat de services een gedefinieerde interface moeten hebben die vastligt in een contract. In het geval van een webservice is dat de WSDL. Ook user interface

services hebben een interface die is vastgelegd in een contract. Op het moment dat de interface is vastgelegd in een contract weten de afnemers wat ze kunnen verwachten van de service. Zo wordt bij webservices onder andere vastgelegd in welke structuur de berichten van en naar de service worden gestuurd.

Voor user interface services is hetzelfde van toepassing. Het interface-contract zal echter niet gaan over een berichtstructuur, maar over de elementen van een user interface die de service ontsluit. Hier speelt de granulariteit een belangrijke rol, in het interface-contract moet worden opgenomen welk granulariteitsniveau de ontsloten componenten van de gebruikersinterface moeten bevatten. Dit kan variëren van een verzameling schermen tot een klein stukje data dat op een bepaalde plaats in de user interface wordt gepresenteerd zoals een veld in een formulier of tabel.

Webservices bestaan in een container – in de meeste gevallen is dit een applicatieserver – waar ze via een fysiek endpoint te benaderen zijn. Voor services van een user interface geldt exact hetzelfde. Zowel webservices als user interface services vertonen in de basis hetzelfde gedrag. Op het moment dat de services beschikbaar zijn en ze ontvangen een bericht/vraag dan stuurt de service een antwoord terug.

Als we bij het realiseren van een user interface dezelfde voordelen willen behalen als bij het opbouwen van een SOA-omgeving, dan is het logisch dat de componenten van een user interface op dezelfde principes moeten zijn gebaseerd als de componenten in een SOA-domein. Alleen de inhoud van de berichten is verschillend. De voordelen van een SOA-omgeving zijn bijvoorbeeld: de herbruikbaarheid van gestandaardiseerde componenten, het gebruik van open standaarden en scheiding van implementatie en interface.

Werkwijze

Een enterprise mashup wordt opgebouwd uit user interface services die naast data en functionaliteit (van een onderliggend systeem) ook een kant-en-klare user interface hebben. Een voorbeeld van een user interface service is een component voor het opvragen van de adresinformatie van een klant. Als inputparameter heeft deze service een klantnummer. De output bestaat uit het adres van de betreffende klant in een user interface-component. De output bevat in dit geval een stuk HTML-code. Deze code toont – geïntegreerd in het grotere geheel van een enterprise mashup – de klantinformatie in verschillende velden op het scherm, zie afbeelding 1. Door middel van bekende HTML-technieken zoals *style sheets* is het mogelijk om het uiterlijk van de getoonde elementen aan te passen aan de stijl die in de enterprise mashup wordt gebruikt.

Met behulp van deze user interface services is het componeren van een enterprise mashup nu zeer eenvoudig. Doordat de verschillende user interface-componenten onafhankelijk

kunnen worden hergebruikt, zijn ze snel te combineren tot nieuwe toepassingen. Het combineren van de componenten van de gebruikersinterface met behulp van de user interface services is echter nog niet voldoende om een volledige enterprise mashup samen te stellen. Een aspect dat nog ontbreekt, is de logica die wordt gebruikt om tot een geïntegreerde mashup te komen. Bij het combineren van verschillende bronnen die zijn gebaseerd op bestaande user interfaces, vormt logica de basis om te weten hoe de mashup zich gedraagt. Dit is ook een van de grote verschillen met de eerder genoemde mashups gericht op de consumentenmarkt. Hierbij ligt het niveau van samenwerking tussen de user interface-componenten veel lager.

Een praktijkvoorbeeld

Als we kijken naar iGoogle zien we dat de verschillende gadgets in de mashup statisch naast elkaar staan in hetzelfde scherm. De interactie tussen deze gadgets is gering. In tegenstelling tot de consumenten-mashups draait het bij een enterprise mashup om het realiseren van interactie met schermcomponenten uit verschillende bronsystemen. De data binnen een gebruikers interface-component zullen vaak bepalend zijn voor de manier waarop data in de mashup worden gebruikt. Een voorbeeld; een verzekeraar maakt gebruik van een ERP-systeem en twee legacy-systemen met daarin polisgegevens. De mashup haalt de klantgegevens op uit het ERP-systeem en combineert deze met de bijbehorende polisgegevens. Het polisnummer uit het ERP-systeem bepaalt welk legacy-systeem er moet worden geraadpleegd om de juiste polisgegevens op te halen. Tevens bepalen de gegevens uit het ERP-systeem welke functionaliteit van toepassing is voor de betreffende klant. Dit houdt in dat de mashup er voor een specifieke klant steeds anders uit zal zien.

Voordelen van de mashup

De mashup is verantwoordelijk voor het sturen van het interactieproces tussen de gebruiker en achterliggende applicaties of bronsystemen. Het gecombineerde gebruik van de applicaties in een enterprise mashup heeft veel voordelen

The screenshot shows a web form titled "Edit An Address". At the top, it says "Please update the fields below to edit your selected delivery address. When you've finished click 'update address'." Below this, there is a red asterisk and the text "These fields must be completed". The form contains several input fields and dropdown menus: "Title" (with a dropdown menu showing "Mr"), "Forename" (with the value "Mike"), "Surname" (with the value "Soer"), "Telephone number" (with the value "01280 84XXXX"), "Telephone type" (with a dropdown menu showing "Home"), "Contact number" (with the value "07977 267260"), and another "Telephone type" (with a dropdown menu showing "Mobile"). There is also a "Reference" field with a text input and a small instruction: "Use this to associate a descriptive reference with the address. e.g. Home." At the bottom, there is a "Make default" checkbox with radio buttons for "Yes" and "No", and a note: "Select yes to make this your default address."

Afbeelding 1: HTML-code toont geïntegreerd in de mashup de klantinformatie.

ten opzichte van het gebruik van de losse applicaties. Waar het voorheen volledig onduidelijk was hoe de diverse bronsystemen naast elkaar worden gebruikt, wordt dit nu ineens afgedwongen door de mashup omdat er nog maar één voorgeschreven en gestandaardiseerde werkwijze is. De tooling waarmee de mashup is gebouwd, meet hoe gebruikers door het proces gaan. Zo kan bijvoorbeeld worden aangetoond dat wordt voldaan aan de vooraf gestelde eisen van gebruik, zoals het aantonen van compliance aan bepaalde regelgeving.

In het praktijkvoorbeeld van de verzekeraar is het mogelijk dat het gebruik van de losse systemen ertoe leidt dat een polis wordt geregistreerd in het tweede polissysteem. Terwijl dit volgens de regels niet zou mogen, omdat in het eerste polissysteem al een vergelijkbare polis aanwezig is. Als deze verzekeraar niet met mashup in de interface werkt, vinden er geen controles plaats bij het invoeren. Hierdoor kan niet worden voorkomen dat een gebruiker die tweede registratie bewerkstelligt. Binnen een enterprise mashup zijn dit soort controles zeer eenvoudig op te nemen, waardoor het afdwingen van compliance of Business Rules over meerdere systemen veel gemakkelijker wordt. Als deze verzekeraar wel gebruik zou maken van een mashup in het ERP-systeem, zou het snel worden opgemerkt dat de klant al een polis heeft. Het invoeren van een nieuwe polis wordt vervolgens onmogelijk gemaakt, doordat de user interface-componenten voor deze functie in de mashup worden weggelaten.

Een enterprise mashup is een snelle manier voor het integreren van onderliggende systemen. Hiermee wordt het aantal mogelijkheden dat beschikbaar is voor integratie met een nieuwe en interessante optie uitgebreid. Een enterprise mashup is een prima mogelijkheid om verschillende (legacy) systemen die niet zijn geïntegreerd, maar waarbij de gegevens wel aan elkaar zijn gerelateerd, op een eenvoudige en snelle manier te integreren. De investering in reeds gerealiseerde user interfaces gaat hierbij niet verloren. De veelal handmatige integratie die gebruikers moeten uitvoeren, wordt bij de enterprise mashup overbodig. Het resultaat is tevens een daling van het aantal fouten dat hierbij wordt gemaakt.

Enterprise mashup tools zoals bijvoorbeeld Corizon staan het toe om zeer dicht bij de business integratie toe te passen, die met de huidige beschikbare technieken veel moeizamer en langzamer mogelijk is. Door het gebruik van herbruikbare, functionele en ook visuele componenten is het voor business gebruikers zeer eenvoudig om te bepalen wat componenten doen en hoe ze kunnen worden gecombineerd. Er zal blijken dat de mashup een bruikbare aanvulling is van het totale op SOA-principes gebaseerde integratiepalet.

Edwin van Asch en Michiel Kemperman zijn respectievelijk Solution Architect en Solution Engineer bij Systemation.