



J-Spring 2007

OSGi: dynamische modules voor Java

Ontwikkelingen in de Java-standaards worden over het algemeen gecontroleerd ingevoerd vanuit een min of meer formeel community process (<http://www.jcp.org>). Soms echter ontstaan er nieuwe Java standaards zonder dat deze vanuit de gereguleerde processen worden bestuurd.

Een mooi voorbeeld van een standaard die buiten de JCP opkomt is OSGi: een module-systeem voor Java. Deze standaard is de enige beschikbare en bewezen standaard op dit gebied is en krijgt steeds meer aandacht bij het grote publiek (voorbeelden hiervan zijn het OSGi plug-in model van Eclipse, de vrijage van Spring met OSGi en de lonkende ogen van een aantal grote JavaEE applicatie-server bouwers). Dit artikel zal proberen iets van de achtergrond en structuur van OSGi te vertellen zodat het makkelijker wordt om alle verhalen beter te kunnen plaatsen.

De OSGi alliance is in 1998 ontstaan, toen enkele mobiele telefonie bedrijven een huis van de toekomst voorzagen waarin allerlei apparaten met elkaar verbonden zouden zijn. De potentiële markt voor applicaties zou zich echter, zo was de gedachte, alleen kunnen ontwikkelen als hiervoor een open standaard beschikbaar zou zijn. Hiertoe werd het Open Services Gateway Initiative opgezet en al snel sloten anderen zich er bij aan. Inmiddels is, om verwarring rond het woord 'gateway' te voorkomen, in de officiële terminologie alleen nog het acronym 'osgi' in gebruik. De OSGi Alliance heeft inmiddels 45 leden met als enkele bekende voorbeelden IBM, BEA, Ericsson en Philips.

De nieuw te ontwerpen standaard moest aan enkele welbepaalde eisen voldoen. Het ging in feite om een heterogeen gedistribueerd embedded netwerk met samenwerkende applicaties, voor de consumentenmarkt. Verschillende applicaties van verschillende leveranciers zouden niet alleen naast elkaar moeten kunnen functioneren maar, en dat was zeer bijzonder, ook met elkaar moeten kunnen samenwerken. Verder moest de gebruiker zo min mogelijk te maken krijgen met allerlei ingewikkelde installatieprocedures, configuratie van instellingen en beheer van versies en aanpassingen. De ontwerpers van de OSGi Alliance die met het ontwikkelen van de specificaties waren belast besloten om hiervoor Java te gebruiken.

Inmiddels in de OSGi standaard geëvolueerd en heeft het het embedded domein verlaten. Gestuwd door het plugin-model van Eclipse en de aandacht uit de enterprise wereld (Spring), blijkt OSGi een veel grotere waarde te hebben dan initieel gedacht. Als je complexe systemen in Java hebt gebouwd weet dat de groeperingsmechanismes in Java (packages, classes en jar-files) nogal tekortschieten. De OSGi standaard is een op services gebaseerd dynamisch



module systeem. De basis voor deze standaard is het OSGi Service Platform. Dit platform levert de primitieven waarmee applicaties opgebouwd kunnen worden op basis van kleine, herbruikbare componenten (zogenaamde bundles). Daarnaast levert het functies waarmee de samenstelling van applicaties dynamisch veranderd kan worden zonder dat de applicatie opnieuw gestart hoeft te worden. Bovenop dit Service Platform heeft de OSGi Alliance een groot aantal standaard interfaces gedefinieerd met functies zoals HTTP servers, configuratie, logging, security, user administratie, XML-parsing en vele anderen.

Meer informatie

Meer informatie kan gevonden worden tijdens vrijwel elk serieus Java-congres. Op de JavaOne, J-Spring en SpringOne wordt OSGi ruim aandacht gegeven. Daarnaast kan de laatste versie van de OSGi specificatie (Release 4) gedownload worden van <http://www.osgi.org/>

Contactpersoon luminis®: dhr. Hans Bossenbroek