Rubriek

OpenESB is een opensource Enterprise Service Bus ontwikkeld in samenwerking met Sun Microsystems. De hoofddoelstelling voor het ontwikkelen van OpenESB is het creëren van een opensource ESB van hoog niveau die kan concurreren met bestaande ESB producten. We geven inzicht in een aantal te verwachten mogelijkheden van OpenESB v3 en vergelijken deze mogelijkheden van de nieuwe versie met een aantal producten welke commercieel op de markt beschikbaar zijn.

OpenESB: een vergelijk

Thijs Volders
is Senior J2EE/SOA Consultant bij Yenlo. Hij is te bereiken via email thijs.volders@yenlo.nl.

Thomas Tromp
is Senior SOA Consultant bij Yenlo. Hij is te bereiken via email thomas.tromp@yenlo.nl.

Sinds 2005, het oprichtingsjaar van het OpenESB project, hebben de ontwikkelingen niet stilgestaan. Op dit moment wordt de laatste hand gelegd aan OpenESB v3 die medio september 2009 geïntroduceerd zal worden. Bezit OpenESB na 4 jaar van ontwikkeling de eigenschappen en kwaliteit om een serieuze kandidaat te zijn bij een ESB toolselectie?

OpenESB en OpenESB.next

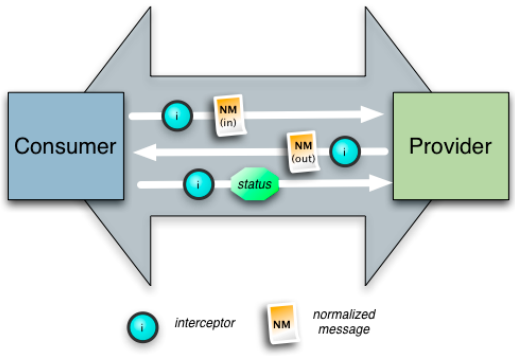
OpenESB.next, ook wel v3 genoemd, is de nieuwe versie van OpenESB. Aan de basis van OpenESB v3 ligt Project Fuji. Dit project is gestart om de ontwikkeling van de nieuwe versie van OpenESB een basis te geven. Deze basis is opgebouwd als OSGi bundle en is gebaseerd op JSR-208 oftewel de Java for Business Integration (JBI) standaard. We beginnen met het uitlichten van een aantal interessante features van OpenESB v3.

OSGi

OpenESB v2 is opgebouwd met in gedachte platformonafhankelijkheid en ontkoppeling van componenten. Veel platformspecifieke code was verborgen achter abstracties en interfaces. Met de komst van OpenESB v3 is dit nog meer verbeterd door de core op OSGi te baseren.
OSGi biedt onder andere een deployment framework waarin deployment artefacten dynamisch en gedeeltelijk geladen kunnen worden. Ook is het mogelijk om verschillende versies van dezelfde classes in geheugen te laden.
Deze manier van classloading brengt complexiteit met zich mee, maar ook veel flexibiliteit. Dat is een reden voor Project Fuji om te kiezen voor OSGi.
Door de core van OpenESB te baseren op OSGi is het mogelijk geworden om deze kleiner dan 300Kb te maken. Daarnaast is het geheugengebruik van OpenESB zo efficiënt mogelijk gemaakt door ervoor te zorgen dat alle componenten, de adapters, interceptors etcetera als OSGi bundle gedeployed kunnen worden. Laad in het geheugen wat nodig is, niet meer, niet minder.
GlassFish v3, de opensource applicatieserver van Sun Microsystems, is een OSGi R4 compliant container, hierdoor is GlassFish een goede keuze om OpenESB op te installeren. Het gebruik van OpenESB verbergt grotendeels het OSGi framework en de afhankelijkheid daarvan voor de ontwikkelaar. Alleen als de ontwikkelaar het OpenESB platform zelf wil uitbreiden is kennis van OSGi nodig. Componenten ontwikkelen voor OpenESB vereist geen OSGi kennis. OpenESB regelt het beschikbaar stellen van de componenten en services als OSGi component.

Interceptors

In OpenESB v3 worden interceptors geïntroduceerd. Een interceptor geeft de mogelijkheid code toe te voegen aan de berichtenstroom tussen aanbieder en klant. Interceptors kunnen gebruikt worden om de hele berichtenuitwisseling, het bericht zelf of zijn payload te onderscheppen en enkele acties hierop te verrichten. Zoals in onderstaand diagram is aangegeven kan elke berichtenstroom bij uitwisseling ondervangen worden.



Figuur 1 Visualisatie Interceptor-implementatie.

Interceptors zijn simpele POJO's en zijn in te zetten om functionaliteiten zoals logging, auditing of

Met de Maven-support is het mogelijk een IDE naar keuze te gebruiken.

beveiliging af te handelen. Ze kunnen door ontwikkelaars in de applicatie of als zelfstandige component gemaakt worden. Doordat in OpenESB v3 interceptors dynamisch (runtime) kunnen worden toegevoegd of verwijderd, kunnen applicaties zelfs na oplevering verrijkt worden met extra functionaliteit.
Interceptors worden ook wel AOP voor SOA genoemd. De werking van interceptors is in grote lijnen vergelijkbaar met AOP. Net als in AOP zijn er pointcuts, join points en advices mogelijk, echter hebben deze wel een andere naam gekregen. De interceptor point is vergelijkbaar met de AOP join point. De interceptor scope is de AOP pointcut en de interceptor met daarbinnen de logica is het advice. Het is mogelijk om meerdere interceptors binnen eenzelfde scope te definiëren. Met behulp van prioriteiten kan de volgorde van deze interceptors worden beïnvloed.
Er is een mogelijkheid interceptors zelfstandig in een eigen OSGI bundle te installeren en onafhankelijk van de applicatie te managen. Hierdoor krijgt de interceptor een eigen lifecycle en is onafhankelijk geworden van de applicatie. Dit is bijvoorbeeld wenselijk wanneer een interceptor bij meerdere applicaties wordt ingezet, zoals bijvoorbeeld bij standaard logging of errorhandling.

Maven support

De ontwikkelaars van OpenESB v3 hebben zoveel mogelijk geprobeerd om vendor lock-in te voorkomen. Zo ook voor de te gebruiken ontwikkelomgeving. Voor OpenESB v3 is een aantal Maven-archetypes beschikbaar waarmee verschillende projecttypen voor Fuji kunnen worden gecreëerd. Ook is er een Fuji-plugin waarmee een deployment artifact gemaakt kan worden. Doordat deze Maven-support is toegevoegd, is het mogelijk om een IDE naar keuze te gebruiken. Maven is prima vanaf de command-line te gebruiken dus ondersteuning vanuit de IDE is niet echt noodzakelijk.

Webbased service composition

Webbased service composition tooling is één van de nieuwe tools die in openESB v3 zal worden geïntroduceerd. Deze tool, een service compositie editor, stelt ontwikkelaars in staat om met een grafische tool de Integration Flow Language (IFL) samen te stellen in een service compositie. De tool kan worden benaderd via een webbrowser.
In plaats van het genereren van een Maven artefact, daarna de IFL te maken, hiervan de properties van de binding components en services aan te passen, kan nu met een grafische drag & drop interface hetzelfde in kortere tijd worden bereikt.
Na het doen van een kort onderzoek blijkt de interface goed bruikbaar. Het draggen & droppen van componenten werkt erg prettig. Door op een component te klikken is het mogelijk om de properties te wijzigen. Ook kunnen door kleine puntjes op componenten koppelingen gelegd worden naar andere componenten in het proces.Een aantal Enterprise Integration Patterns is standaard beschikbaar waarmee bijvoorbeeld berichten gebroadcast kunnen worden naar meerdere componenten.

Integration Flow Language (IFL)

IFL is een domain specific language om routing (de flow) tussen services te definiëren. Veel input van de community is gebruikt om deze taal te maken tot wat het nu is geworden. Een doel van IFL is om in een natuurlijke, simpele taal de berichtenstroom tussen services te realiseren. Kijkend naar de syntax van IFL lijkt dit doel te zijn bereikt. Een IFL bestand bestaat uit de definitie van een aantal services en één of meer routing definities.
Services worden gedefinieerd als servicetype met een identifier. De routing definities beginnen met het keyword route en kunnen eventueel een nesting van andere keywords bevatten. Er is zoveel mogelijk ruis uit de taal gehouden om de eenvoud te behouden.
Er zijn ook een aantal zaken die (nog) niet in IFL zijn geïmplementeerd. Zo is het niet mogelijk om operaties op services aan te roepen, evenals namespacing van services. Later worden deze misschien geïmplementeerd.
Er wordt op dit moment gewerkt aan ondersteuning voor standaard Message Exchange Patterns. Hiermee kan men aangeven of services fire-and-forget berichten krijgen of dat er een request-response bericht wordt gecommuniceerd.

Vergelijkend met bijvoorbeeld SCALA DSL is de syntax simpel en duidelijk.

Apache Camel DSL:

```
public class SimpleRouteBuilder extends RouteBuilder {
 public void configure() {
    from("queue:foo").choice().when(xpath("//foo").
isEqualTo("abc"))to("queue:bar").otherwise().
to("queue:others");
 }
}
```

IFL:

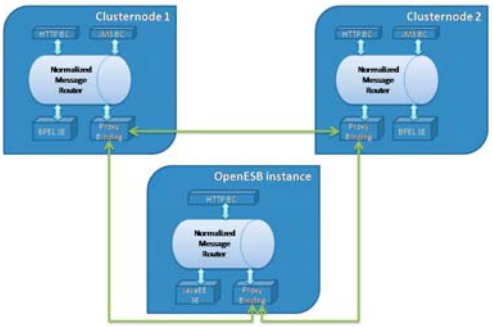
```
file "foo"
file "others"

route do
    from "foo"
    select xpath "//foo" do
       when "abc" route to "foo"
       else route to "others"
    end
end
```

In dit voorbeeld is te zien dat java-code niet aan de orde is. De IFL wordt in een eigen bestandstype opgeslagen waardoor syntaxhighlighting en

formatteringregels specifiek voor dit bestandstype kunnen worden toegepast. Door DSL in javacode te verwerken, zoals in het voorbeeld hierboven van toepassing is bij het gebruik van Apache Camel, is de structuur minder overzichtelijk. De geneste formattering wordt teniet gedaan door de formatter van de editor waarin gewerkt wordt.

Ondersteuning voor een gedistribueerde Service Bus.

In OpenESB v3 is een proxy binding component geïntroduceerd. Dit component maakt het mogelijk om meerdere (verschillende) ESBs met elkaar te verbinden. De ESB is als het ware uitgebreid via een netwerk.
Figuur 2 toont een gedistribueerde JBI bestaande uit drie instanties. De proxy bindings communiceren met elkaar om informatie uit te wisselen over beschikbare endpoints en om gegevens over te brengen naar de andere instanties.



Figuur 2: Gedistribueerde ESB.

Vergelijk nieuwe features met concurrerende producten

OpenESB is niet het enige beschikbare ESB-product op de markt. Kijkend naar de implementatie van de techniek ziet het er veelbelovend uit. Maar hoe staat het met andere ESB's? Bieden zij features die vergelijkbaar zijn met die van deze nieuwe OpenESB release? Het gaat nu te ver om alle features van OpenESB met andere producten te vergelijken, eveneens moeten we een selectie maken uit de beschikbare ESB's. Er is gekozen om de eerder in dit artikel behandelde nieuwe features van OpenESB te vergelijken met de op dit moment veel gekozen ESB-producten, te weten: IBM WebSphere Enterprise Service Bus, BEA AquaLogic en Oracle SOA Suite.

N.B: De feature IFL is door het Fuji project geïnitieerd. Door het specifieke karakter van deze feature is deze niet in het vergelijk op te nemen.

IBM

De IBM WebSphere software suite omvat veel verschillende producten waaronder WebSphere Enterprise Service Bus en IBM WebSphere Message Broker. Deze twee producten zijn het meest herkenbaar als onderdeel van de ESB software stack. Zowel een kracht als een zwakte in de IBM WebSphere software stack is de keuzemogelijkheid. Er zijn heel veel verschillende producten die allemaal iets met ESB te maken hebben. Om erachter te komen wat deze producten bieden is een studie op zich noodzakelijk.

De IBM WebSphere ESB bestaat uit WebSphere Application Server (WAS) waarop WebSphere ESB (WESB) geïnstalleerd kan worden. Evenals OpenESB is WAS en WESB een OSGi bundle. IBM WebSphere is gebaseerd op Equinox, de Eclipse OSGI R4 core framework implementatie.
In tegenstelling tot Sun GlassFish applicatie server is dus WebSphere zelf geen OSGi framework implementatie maar is deze als bundle op Equinox beschikbaar gesteld. Omdat IBM backwards-compatibility wil blijven garanderen tot op J2EE 1.2 niveau, is het niet mogelijk om OSGi bundles te deployen op WebSphere. Ontwikkelaars zijn hierdoor nog aangewezen op het traditionele deployment model.
In WebSphere ESB ontbreekt ondersteuning voor message-interceptors. Het injecteren van functionaliteit in bestaande berichtenstromen is niet zomaar mogelijk. Dit betekent dat de applicatie moet worden aangepast om de berichtenstroom aan te passen waarna deze opnieuw kan worden deployed.
Maven support voor WebSphere is wel aanwezig. Maven heeft plugins voor de WebSphere Application Server. Vanwege het traditionele deployment model zijn de standaard Maven plugins voor WAR/EAR en JARs voor handen en kunnen dan ook gebruikt worden.
Message routing wordt in WebSphere gerealiseerd met behulp van mediation modules. Deze kunnen deployed worden in de ESB en worden gemaakt met behulp van een flow editor in WebSphere Integration Developer.
WebSphere ESB heeft ondersteuning voor een geclusterde omgeving. IBM heeft zich altijd op de enterprise gericht waarin clustering veelal wordt toegepast. WebSphere maakt gebruik van Service repositories en registries waarmee de locatie van de verschillende services kan worden bepaald. De ESB zorgt ervoor dat berichten op een andere ESB terecht komen indien de service zich buiten de lokale ESB bevindt.

Oracle en BEA Systems

Doordat Oracle het afgelopen jaar BEA heeft overgenomen, heeft het concern het ESB-marktaandeel drastisch vergroot.
Oracle heeft plannen om de Oracle ESB en de AquaLogic service bus samen te voegen tot één product. Dit vergelijk gaat nog uit van twee gescheiden

Met de proxy binding component kun je verschillende ESB's verbinden.

	OSGi	Interceptors	Maven support	Webbased service composition	Integration Flow Language (IFL)	Gedistribueerde Service Bus	Vender lockin Prijs
OpenESB v3	✓	✓	✓	✓	✓	✓	✗
IBM WebSphere	✗*	✗	✓	✓	✗	✓	✓
Oracle ESB	✓	✓	✓	✗	✗	✓	✓
AquaLogic Service Bus	✗*	✓	✓	✓	✗	✓	✓

(*) alleen via Equinox

Figuur 3: Overzicht features ESB producten.

producten omdat nog niet geheel helder is welke eigenschappen in welke hoedanigheid in het nieuwe ESB product worden opgenomen.
In de komende Oracle Fusion Middleware 11g release wordt OSGi ondersteund. BEA AquaLogic was voor de overname al voorzien van OSGi ondersteuning in de vorm van Equinox als micro-kernel. In deze vorm is AquaLogic evenals WebSphere geen OSGi framework maar is deze via Equinox beschikbaar.
Oracle SOA Suite en BEA AquaLogic hebben beide een mogelijkheid voor message-interceptors die nagenoeg eenzelfde functionaliteit bieden als OpenESB. Beide producten hebben ook Maven ondersteuning.
Voor het verschijnen van AquaLogic versie 3.0 werden alle ESB werkzaamheden Web-based uitgevoerd, AquaLogic ondersteunde daarmee het web-based werken al enige tijd. AquaLogic heeft echter sinds versie 3 naast de web based tool ook de Eclipse ondersteuning. Oracle biedt geen web-based compositite tool, maar biedt wel de ESB web console waar verschillende zaken bekeken en aangepast kunnen worden. JDeveloper is tot op heden de tool waarmee in de SOA Suite de ESB wordt opgezet.
De AquaLogic Service Bus geeft via een zogenoemde proxy service de mogelijkheid om via de Service Bus Console een message flow te configureren. Via JDeveloper heeft Oracle ESB ook messageflow ondersteuning. Dit gebeurt via een grafische interface waarmee message flows opgebouwd kunnen worden.
Zowel AquaLogic Service Bus als Oracle ESB hebben de mogelijkheid om geclusterd te opereren. Hiermee hebben ook deze producten de voordelen van load-balancing en schaalbaarheid.

OpenESB nadelen

OpenESB is goed op weg, echter ontbreekt er aan het product nog een standaard ready-to-use console van waaruit de ESB gemakkelijk te monitoren en managen valt.
Via de zogenaamde 'Management en Monitoring API' is er echter wel een mogelijkheid om ESB informatie, waaronder ESB runtime infomatie, te ontsluiten. Helaas is het via deze weg wenselijk om een front-end applicatie te schrijven die de onttrokken informatie inzichtelijk maakt voor gebruikers.
De OpenESB community heeft het gebrek aan een ESB console onderkend. Ten tijde van dit schrijven wordt er gewerkt aan de OpenESB console.

Oracle en OpenESB

Sinds de overname van Sun door Oracle is er weinig bekendgemaakt over de toekomst van SUN's open source software(OSS) producten. Op vele fora wordt hier heftig over gediscussieerd en de algemene aanname is dat Oracle de doorontwikkelde producten, met uitzondering van MySQL, niet zal laten vallen. Door deze overname heeft Oracle een geweldige positie ingenomen ten opzichte van OSS. Het is dan ook een uitgelezen moment om haar standpunt kracht bij te zetten door de verkregen OSS producten verder te ontwikkelen. Onze verwachting is dat de ontwikkeling van een dergelijk krachtig en volwassen product als OpenESB wordt voortgezet en er nog vele releases zullen volgen.

Conclusie

De in dit artikel behandelde features zijn slechts een greep uit de totale lijst van features van OpenESB v3. OpenESB is inmiddels uitgegroeid tot een volwassen alternatief voor de bestaande ESB producten op de markt. Door support van de community zijn een hoop obstakels en probleempunten uit bestaande producten bekeken en is hiervoor een oplossing geïntroduceerd.
Doordat er op alle niveaus mogelijkheden zijn om zonder problemen andere frameworks te gebruiken dan degene die de ontwikkelaars van OpenESB hebben gekozen, biedt het platform veel keuzevrijheid. OpenESB kan op verschillende applicatieserver-producten die in de markt beschikbaar zijn worden geïnstalleerd.
De vergelijking met drie alternatieven in de markt toont aan dat ook deze producten goed mee kunnen komen met de ontwikkelingen, echter tegen een prijs. Een niet te onderschatten voordeel van OpenESB ten opzicht van de andere producten uit dit artikel is het feit dat het opensource is en dat er geen licentiekosten aan verbonden zijn.
Vendor lock-in is meestal niet te voorkomen bij een keuze voor één van de drie andere producten doordat er proprietary interfaces worden gebruikt of doordat er extensie op bestaande standaarden worden geleverd welke niet portable naar andere vendor producten zijn.
Dit alles maakt dat OpenESB v3 zeker niet onderschat moet worden ten tijde van een software selectieproces. Dit product zou op elke longlist een plaatsje moeten krijgen; er bestaat een erg grote kans dat dit product ook de shortlist zal halen of zelfs zal domineren. «

OpenESB is een volwassen alternatief voor bestaande producten.

Referenties

- Voor meer informatie over deze nieuwe release van OpenESB kijk op open-esb.dev.java.net en www.GlassFish.nl.
- http://www.oracle.com/technology/tech/opensource/oracle-open-source-faq.html