Orientation in Objects

Die SOAP Engine Apache AXIS

oder: Die Achse des Guten

) Schulung)

) Beratung)

) Entwicklung)

AUTOR

Thomas Bayer
Orientation in Objects GmbH

Veröffentlicht am: 27.2.2003

ABSTRACT

Web Services sind das Bindeglied zwischen der Java Welt und Microsofts .NET. SOAP ist zu einem Standard für die Kommunikation zwischen verschiedenen Sprachen und Plattformen geworden. Dieser Artikel beschreibt das SOAP Werkzeug Apache AXIS.

) Artikel)

Orientation in Objects GmbH

Weinheimer Str. 68
D-68309 Mannheim

Tel. +49 (0) 6 21 - 7 18 39 - 0
Fax +49 (0) 6 21 - 7 18 39 - 50

www.oio.de info@oio.de

ÜBERBLICK

Apache AXIS ist eine Open Source Implementierung des Web Service Standards SOAP unter der Lizenz der Apache Software Foundation. AXIS kann für die Entwicklung von Clients, Servern und Gateways verwendet werden. Das Hosting von Web Services, d.h. der Betrieb und das Anbieten von Web Services kann ebenfalls mit AXIS durchgeführt werden. AXIS bietet die folgenden Highlights:

- Open Source unter der Lizenz der Apache Software Foundation
- Hohe Performanz
- Flexible Konfiguration und Erweiterbarkeit
- Unabhängigkeit vom Transport. Es kann HTTP, FTP oder Mail verwendet werden.
- Unterstützung von SOAP 1.1 und teilweise von 1.2
- WSDL Unterstützung über Code Generatoren
- JAX-RPC und SAAJ Konformität
- Erweiterungen für Sicherheit
- Provider für den Zugriff auf Enterprise JavaBeans
- ...

ENTSTEHUNG

AXIS ist ein Nachfolgeprojekt der beliebten Apache SOAP Implementierung, die aus dem IBM Projekt SOAP4J hervorging. Die Architektur von AXIS wurde von Grund auf neu gestaltet. Apache SOAP basiert noch auf dem langsamen Document Object Model DOM. AXIS verwendet dagegen SAX Filterketten. In Punkto Funktionalität, Performanz und Interoperabilität hat AXIS ab der Version 1.0 die Apache SOAP Implementierung bereits überrundet. Ein wesentlicher Vorteil von AXIS ist die konsequente Ausrichtung an die JAX-RPC Spezifikation von Sun.

Für AXIS wurde nicht der Name Apache SOAP Version 3.0 verwendet, da zum damaligen Zeitpunkt eine Unsicherheit über die Bezeichnung der SOAP Technologie beim W3C bestand. Der Name AXIS wurde mit der Bedeutung *Apache eXtensible Interaction System* möglichst nichtssagend gehalten.

ARCHITEKTUR

AXIS ist keine starre Software aus einem Guss. Module gliedern AXIS in Subsysteme, die verschiedene Aufgaben übernehmen. Die Verarbeitung von Anfragen erfolgt über Filterketten, die sich aus Handlern zusammensetzen.

Die Verkettung der Handler ist flexibel und kann über Konfigurationsdateien angepaßt werden. Über eigene Handler kann in die Verarbeitung von SOAP Nachrichten eingegriffen werden. Typische Aufgaben für die Handler sind Logging oder Sicherheitsüberprüfungen.

Bei der Anwendungsentwicklung ist es nicht notwendig selbst in die Filterketten einzugreifen. Mit dem Konzept der Ketten ist AXIS jedoch offen für Erweiterungen.

Damit Web Services bereit gestellt werden können, wird ein Server benötigt, der Web Anfragen verarbeiten kann. AXIS kann in einen Web Container wie den Tomcat integriert werden. Es gibt dafür eine Web Anwendung, die AXIS in Form von drei Servlets beinhaltet. Abbildung 1 zeigt einen Web Container, in welchem die drei Servlets installiert wurden.

Das AXIS Servlet kann Web Services aufnehmen und Anfragen an diese zur Weiterverarbeitung routen. Aufgrund der Zustellungsfunktion spricht man von einem SOAP Router oder Dispatcher.

Das Admin Servlet dient der Fernwartung des SOAP Routers. Web Services können über dieses Servlet installiert und deinstalliert werden.

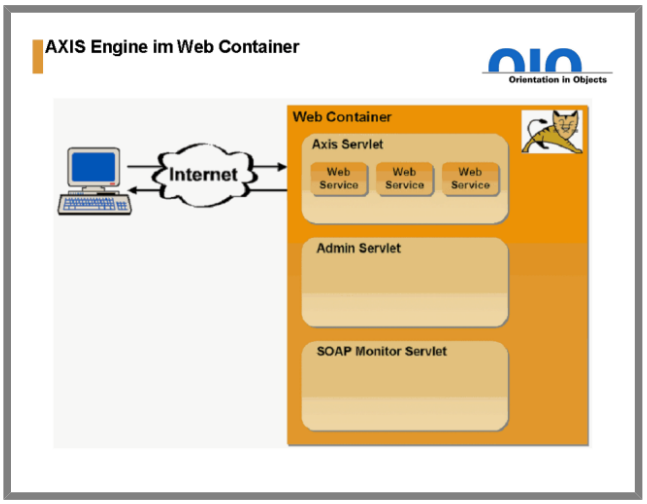

Abbildung 1: AXIS Engine im Web Container

WAS IST DEPLOYMENT?

Um einen Web Service zu nutzen muß dieser zuvor im SOAP Router installiert werden. Man bezeichnet diesen Vorgang als Deployment. Ein installierter Web Service hat zur Laufzeit bestimmte Eigenschaften, die während des Deployments festgelegt werden können. In einem Deploymentdescriptor können diese Eigenschaften festgelegt werden. Für den Deploymentdescriptor wird eine XML Datei verwendet.

SAAJ UND JAX-RPC KONFORMITÄT

Die Verwendung von SOAP basierten Web Services mit Java wird mit Spezifikationen und Schnittstellen beschrieben. JAX-RPC definiert den Aufruf von entfernten Methoden während SAAJ den Zugriff auf SOAP mit Attachments regelt.

Nach dem Bestehen der *Compliance Tests* für JAX-RPC und SAAJ von Sun wurde die Version 1.0 von AXIS freigegeben.

WAS IST SAAJ?

SAAJ oder *SOAP with Attachments API for Java* ist eine Spezifikation von Sun. AXIS implementiert die Schnittstellen dieser Spezifikation und verwendet SAAJ als Basis.

WAS IST JAX-RPC?

JAX-RPC beschreibt Schnittstellen und Konventionen für entfernte Funktionsaufrufe. Das Mapping von Java Datentypen auf XML Schema Typen, die von SOAP verwendet werden, wird beschrieben. Für die Generierung von Java Klassen aus WSDL Dateien und die Erzeugung von WSDL aus Java Klassen gibt es Vorschriften.

INTEROPERABILITÄT

Wichtig für die Verbreitung von Web Services ist die Interoperabilität zwischen den verschiedenen Tools und Plattformen. Ein interoperabler Web Service kann ohne spezielle Anpassungen von anderen Plattformen benutzt werden. Beispielsweise kann ein mit Java realisierter Web Service von .NET Clients verwendet werden.

Zu welchen C++, Perl, Python, etc. SOAP Implementierungen AXIS kompatibel ist, kann dem SOAP Builder Vergleich entnommen werden. Dort sind Testergebnisse aufgeführt, die bis zu den einzelnen Datentypen die Interoperabilität zwischen den Tools auflisten.

WSDL UNTERSTÜTZUNG

AXIS kümmert sich um die Codierung und Decodierung von XML Nachrichten. Der Entwickler kann sich ganz auf seine Anwendung konzentrieren.

WSDL oder die *Web Services Description Language* beschreibt einen Web Service so, dass mit den Informationen dieser Beschreibung eine komfortable Verwendung eines Web Service möglich wird. Neben der Adresse des Servers enthält WSDL auch eine Beschreibung der Funktionen und der zugehörigen Daten für einen Aufruf.

WSDL ist nicht auf eine Programmiersprache oder eine Plattform begrenzt. Eine WSDL Beschreibung eines mit C# realisierten .NET Web Service kann für die Erzeugung eines passenden Clients mit Java verwendet werden.

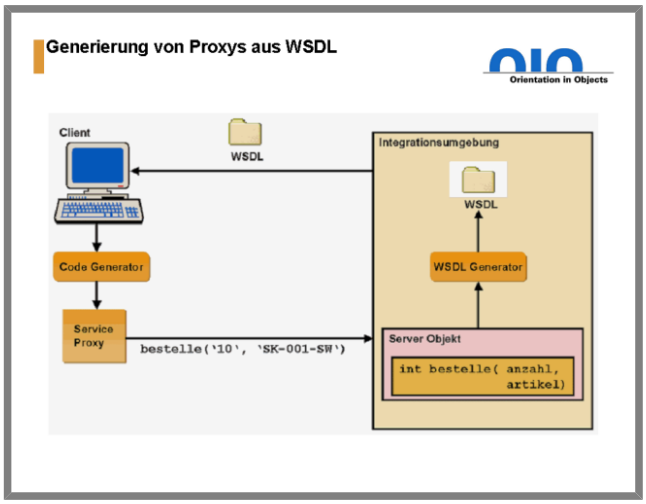

Abbildung 2: Generierung von Proxys aus WSDL

AUTOMATISCHE WSDL GENERIERUNG

WSDL Dateien können für installierte Web Services abgefragt werden. Die WSDL Beschreibung des Web Services wird automatisiert erzeugt. Zum Abfragen reicht ein einfacher Web Browser.

ERZEUGEN VON CLIENT KLASSEN AUS WSDL

Aus einer WSDL Beschreibung eines Web Services können mit dem Tool *WSDL2Java* Klassen für Client und Server erzeugt werden. Für den Server werden Rumpfklassen erzeugt, mit deren Hilfe sich leicht Serverfunktionalität ansprechen läßt. Mit den Client Klassen kann mit wenigen Zeilen Code ein Client für einen bestehenden Web Service erstellt werden.

ERZEUGEN VON WSDL AUS JAVA CODE

WSDL Dateien sind recht kompliziert aufgebaut und eignen sich nicht für eine Erstellung von Hand. Das AXIS Werkzeug *Java2WSDL* kann aus normalen Java Klassen eine WSDL Datei generieren. Die WSDL Datei kann über einen Web Server veröffentlicht werden, um Fremden die Nutzung eines Web Services zu erleichtern.

Ausgehend von einer WSDL Datei kann mit einer leistungsfähigen Entwicklungsumgebung in Minuten ein Client für einen unbekannten Web Service erzeugt werden.

ANFORDERUNGEN

Für eine AXIS Installation wird Java ab der Version 1.3 und ein Servlet 2.2 konformer Web Container benötigt. Die Entwickler von AXIS empfehlen die Verwendung von Tomcat.

SICHERHEIT

AXIS bietet die Möglichkeit die Client Anfrage und die Antwort vom Server mit SSL zu verschlüsseln. Darüber hinaus kann AXIS mit dem XML-Security Projekt von Apache eingesetzt werden, um Teile einer SOAP Nachricht zu verschlüsseln. So kann der Body einer Nachricht, der die Nutzdaten enthält verschlüsselt werden, während der Header unverschlüsselt bleibt, um ein Routing zu ermöglichen. Mit AXIS wird ein Codebeispiel ausgeliefert, welches den Einsatz von XML-Security für die Signierung des SOAP Umschlages demonstriert.

ZUGRIFF AUF ENTERPRISE JAVABEANS

Methoden eines stateless SessionBeans können als Web Services angeboten werden, ohne eine Zeile Code zu schreiben. Es genügt ein Deployment Descriptor, der den Web Service beschreibt. Im Descriptor wird der Name des Beans und seine Schnittstellen angegeben. Mit dem Deployment Descriptor kann ein Web Service installiert werden, der Zugriff auf das Enterprise JavaBean ermöglicht.

Abbildung 3 zeigt wie EJBs als Web Service angeboten werden können. Der Application Server mit den Enterprise JavaBeans kann sich sogar auf einem anderen Rechner befinden, als der Web Service. Schickt der Client eine Anfrage an den Web Service, wird diese entgegengenommen und an den Application Server delegiert. Für den Application Server ist es ein Aufruf wie jeder andere auch. Er verarbeitet die Anfrage und liefert einen Rückgabewert, der an den Web Service geschickt wird. Der Web Service verpackt den Wert in eine SOAP Nachricht im XML Format und sendet diese zurück an seinen Client.

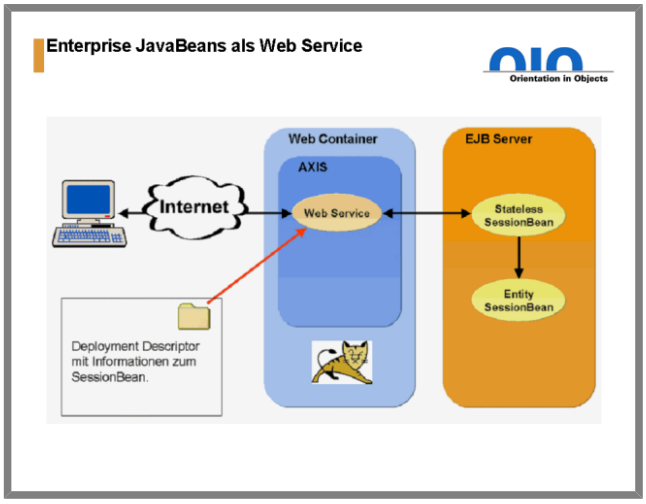

Abbildung 3: Enterprise JavaBeans als Web Service

TOOLS

Neben den WSDL Generatoren sind im AXIS Paket weitere Werkzeuge für den Entwickler enthalten.

MONITORING

Der TCP Monitor kann zwischen Client und Server geschaltet werden. SOAP Nachrichten, die zwischen Server und Client ausgetauscht werden, können über eine graphische Oberfläche verfolgt werden. Die ausgetauschten Nachrichten sind dank XML sowohl für Maschinen als auch Menschen lesbar. Der Monitor ist für Debugging und für die Suche nach Interoperabilitätsproblemen hilfreich.

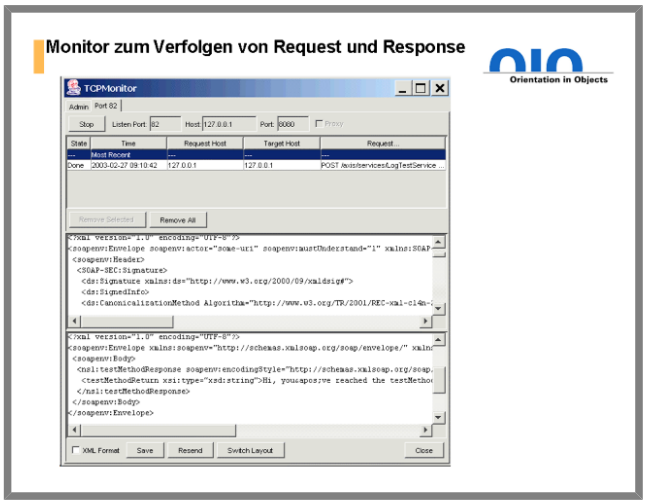

Abbildung 4: Monitor zum Verfolgen von Request und Response

ZUKUNFT VON AXIS

Teile der kommenden SOAP 1.2 Spezifikation sind bereits umgesetzt. Funktionserweiterungen und Verbesserungen der Architektur sind geplant. C++ Entwickler können sich auf eine kommende C++ Version für den Client freuen.

Die W3C Standards SOAP 1.1 und in Teilen die kommende SOAP Version 1.2 sowie WSDL werden von AXIS wie die Java Standards JAX-RPC und SAAJ unterstützt.

Werkzeuge wie die WSDL Generatoren und schnelles Deployment über das Admin Interface verhelfen dem Entwickler zu mehr Produktivität bei der Entwicklung von Web Services. Detailierte Kenntnisse der SOAP Spezifikation sind für das Verwenden und die Erstellung von Web Services nicht mehr notwendig.

AXIS ist ein würdiger Nachfolger von Apache SOAP, der mit vielen Features, einer ausgereiften Architektur und mit Standardkonformität glänzen kann. AXIS bietet mit seiner ausgereiften Architektur und der Open Source Lizenz eine gute Basis für Anwendung und Web Service Tools.

BIBLIOGRAPHIE

- AXIS bei der Apache Software Foundation
  (http://ws.apache.org/axis/)
- Sam Ruby's Axis/SOAP interop page
  (http://www.apache.org/~rubys/ApacheClientInterop.html)
- XMethods
  (http://www.xmethods.org/)