

Een evolutionaire stap voorwaarts

Webservices nieuwe fase in internetontwikkeling

Webservices staan zeer in de belangstelling. Grote leveranciers als Hewlett Packard, IBM, Microsoft, Sun en Oracle hechten er groot belang aan om het ontwikkelen en beschikbaar stellen van webservices te ondersteunen op hun platformen. Webservices worden door velen gezien als voorwaarde voor de ontwikkeling van de derde generatie Internetapplicaties. In een tweetal artikelen zal Erwin Groenendal uiteenzetten wat webservices precies zijn, welke standaards een rol spelen, welke toepassingen er zijn en welke ondersteuning Oracle levert op dit gebied.

In dit eerste artikel geeft Groenendal een definitie van een webservice en gaat hij in op de toepassingen en standaards. In het tweede deel, dat in de volgende Optimize zal verschijnen, wordt meer gedetailleerd ingegaan op de standaards en wordt de ondersteuning van webservices in Oracle⁹ⁱAS en JDeveloper⁹ⁱ uitgebreid behandeld. In het decembernummer van Optimize zei Hans Bos, marketing manager bij Oracle Nederland: "Oracle's strategie is de komende jaren vooral gericht op webservices". Dergelijke uitingen bewijzen dat Oracle groot belang hecht aan webservices.

Definitie

De genoemde grote leveranciers en een aantal kleine gespecialiseerde leveranciers als Avinon, Iona en Bowstreet hanteren allemaal definities voor webservices die doorspekt zijn met prachtige Amerikaanse marketingtermen. Hoewel de definities enigszins verschillen zijn er een vijftal elementen in de definities aan te wijzen die telkens terugkeren en daarmee goed aangeven wat een webservice in feite is.

Software component

Een webservice is een software component. Denk hierbij aan een PL/SQL procedure of functie of een Java method.

Eenduidige functionaliteit

Een webservice biedt eenduidige functionaliteit. Dit houdt twee dingen in: ten eerste dat de business logica die de software

component implementeert duidelijk is en ten tweede dat de data types van de parameters en return waarden bekend zijn.

Gedistribueerd

De aanroep van de webservice vindt gedistribueerd plaats. De webservice bevindt zich op een ander systeem dan waarvandaan de aanroep gedaan wordt.

Internet standaards

Voor het aanroepen, beschrijven en registreren van webservices wordt gebruik gemaakt van Internet standaards. Deze standaards betreffen in ieder geval de algemene standaards HTTP en XML, maar ook de specifieke webservices standaards SOAP, WSDL en UDDI. Voor de aanroep van een webservice wordt gebruik gemaakt van HTTP, XML en SOAP, voor de beschrijving van WSDL en voor het registreren van UDDI. Deze standaards worden verderop in dit artikel verder toegelicht.

Zelfbeschrijvend

Een webservice kan zelfbeschrijvend worden genoemd omdat met behulp van WSDL (Webservices Description Language) de webservice precies beschreven kan worden. Dit houdt onder andere in dat de datatypes van de parameters en return waarden precies gedefinieerd worden met behulp van de XML Schema standaard.

Samenvattend komen we dan uit op de volgende definitie van een webservice:

Een webservice is een software component die eenduidig zelfbeschreven functionaliteit biedt en gedistribueerd aangeroepen wordt door gebruik te maken van standaard Internet technologie.

Aan deze definitie moet nog de opmerking worden toegevoegd dat een webservice zélf dus géén user interface biedt. De web-

service wordt aangeroepen vanuit een applicatie. Deze applicatie kan een user interface bieden, maar de webservice zelf doet dit niet. Een HTML pagina waarmee bepaalde informatie kan worden opgevraagd is géén webservice, maar kan wel, “onder de kap” gebruik maken van een webservice om de aangeboden functionaliteit te implementeren.

Toepassingen van webservices

Nu we gedefinieerd hebben wat een webservice precies is kunnen we gaan kijken naar de toepassingen van webservices. Hierbij maken we onderscheid tussen de partij die de webservice beschikbaar stelt, de “provider”, en de partij die de webservice afneemt, de “consumer”. De toepassingen van webservices zijn te categoriseren aan de hand van een viertal scenario's.

Business-to-Business (B2B)

In dit scenario stelt een bedrijf webservices beschikbaar aan haar business partners om gestructureerde informatie op te vragen en eenvoudige transacties uit te voeren. De business partner kan de webservices aanroepen vanuit haar eigen applicaties om zodoende bedrijfsprocessen te integreren. De provider van de webservices bindt hiermee de business partner, de consumer van de webservices, aan zich en bespaart kosten door het efficiënt automatiseren van deze processen. De consumer bespaart eveneens kosten en kan een betere service bieden aan haar klanten. Een goed voorbeeld hiervan is een vervoerder die een webservice beschikbaar stelt voor het opvragen van de status van bezorgopdrachten. Een bedrijf dat gebruik maakt van de diensten van de vervoerder voor het bezorgen van goederen kan de webservice gebruiken om op haar eigen website inzicht te geven in de status van de bezorging aan haar klanten.

Application Service Provider (ASP)

Een bedrijf dat applicaties host, biedt in dit scenario haar klanten de mogelijkheid om andere systemen te koppelen met de gehoste applicatie op basis van webservices. De Application Service Provider breidt hiermee haar product portfolio uit door haar klanten niet alleen toegang te bieden tot de gehoste applicatie via de standaard user interface, maar ook tot essentiële koppelmogelijkheden met niet-gehoste systemen. De klant van de ASP kan op deze manier haar bedrijfsprocessen optimaliseren en kosten besparen. De Oracle e-business suite is hier een goed voorbeeld van. Deze applicaties bevatten webservices voor de koppeling met andere systemen.

Content Syndication

Bedrijven die content zoals nieuws-, weer- en fileberichten en aandelenkoersen aanbieden aan portals en andere websites kunnen webservices gebruiken om de content aan te leveren. In dit scenario is er sprake van een nieuw afzetkanaal voor de leverancier van de content in toevoeging op het aanleveren van

de content via bijvoorbeeld FTP of e-mail. De afnemer van de content krijgt mogelijkheden om op een eenvoudigere manier (gepersonaliseerde) content te plaatsen op haar portal of web site en op deze manier consumenten beter aan zich te binden.

Application-to-Application (A2A)

In dit laatste scenario worden webservices gebruikt voor het beschikbaar stellen van een softwarecomponent voor gebruik vanuit meerdere interne applicaties. Doordat applicaties vaak op verschillende platformen draaien en in verschillende programmeertalen ontwikkeld zijn is het complex en kostbaar om gemeenschappelijk componenten of modules te ontwikkelen die door al deze applicaties gebruikt kunnen worden. Dit komt met name doordat de verschillende platformen verschillende technologieën hebben voor gedistribueerde verwerking (DCOM, CORBA, EJB, et cetera). Door gebruik te maken van webservices is het veel eenvoudiger en goedkoper om hergebruik van software componenten te realiseren.

Uit bovenstaande scenario's kunnen we afleiden dat webservices zowel kosten kunnen besparen als ook omzet kunnen verhogen. De omzetverhogende voordelen betreffen het binden van business partners en klanten, het aan kunnen bieden van nieuwe producten en het benutten van nieuwe afzetkanalen. Kosten kunnen bespaard worden door het hergebruik van software componenten, het optimaliseren van bedrijfsprocessen en de flexibiliteit en eenvoud van integratie.

Revolutie of evolutie?

Webservices worden vaak gepositioneerd als een technologie die revolutionaire toepassingen mogelijk maakt. Denk dan bijvoorbeeld aan auto's, die geautomatiseerd een afspraak maken met de garage voor een onderhoudsbeurt en de afspraak in de elektronische agenda van de bestuurder plaatsen. Of die, als er getankt moet worden, pompstations in de buurt laat bieden op het leveren van de benodigde brandstof, vervolgens een keuze maken op basis van de prijs, afstand, verkeerssituatie en andere factoren om daarna via het navigatiesysteem de bestuurder naar het pompstation te leiden. Dit soort revolutionaire toepassingen zullen vroeg of laat komen. Webservices kunnen gebruikt worden om (delen van) deze toepassingen te realiseren. Als technologie zijn webservices echter geen revolutie maar een evolutie; en wel in drie opzichten.

Distributed computing

Na platformspecifieke technologieën voor gedistribueerde verwerking als DCOM, CORBA en EJB met bijbehorende “uitdagingen” om deze op elkaar aan te laten sluiten vullen Webservices, in het bijzonder SOAP (Simple Object Access Protocol), de behoefte in om op een eenvoudige manier gedistribueerde verwerking te realiseren.

Niet langer interactief

In de eerste fase van het Internet hebben bedrijven vooral toepassingen beschikbaar gesteld waarmee op een interactieve manier gestructureerde informatie kan worden opgevraagd en eenvoudige transacties kunnen worden uitgevoerd. In business-to-business situaties is echter in toenemende mate de behoefte ontstaan om het, omslachtige, interactieve aspect weg te nemen en systemen direct met elkaar te integreren.

Eenvoudige B2B en A2A integratie

Voor het integreren van bedrijfsprocessen tussen bedrijven (B2B) en tussen applicaties binnen bedrijven (A2A of EAI) is het gebruik van XML (na het EDI tijdperk) de algemeen geaccepteerde standaard. Veelal wordt hier (asynchrone) message queuing technologie voor gebruikt. Hoewel deze technologie voor complexe transacties en in bepaalde situaties nodig is, is er ook behoefte aan een eenvoudige manier om B2B en A2A integratie te realiseren. Webservices vullen deze behoefte uitstekend in.

Standards

We zijn nu op het punt gekomen om eens op een rijtje te zetten welke standards een rol spelen bij webservices.

HTTP

Het Hypertext Transfer Protocol wordt door een groot deel van de wereldbevolking (onbewust) dagelijks gebruikt voor het benaderen van web pagina's op het Internet. HTTP wordt ook

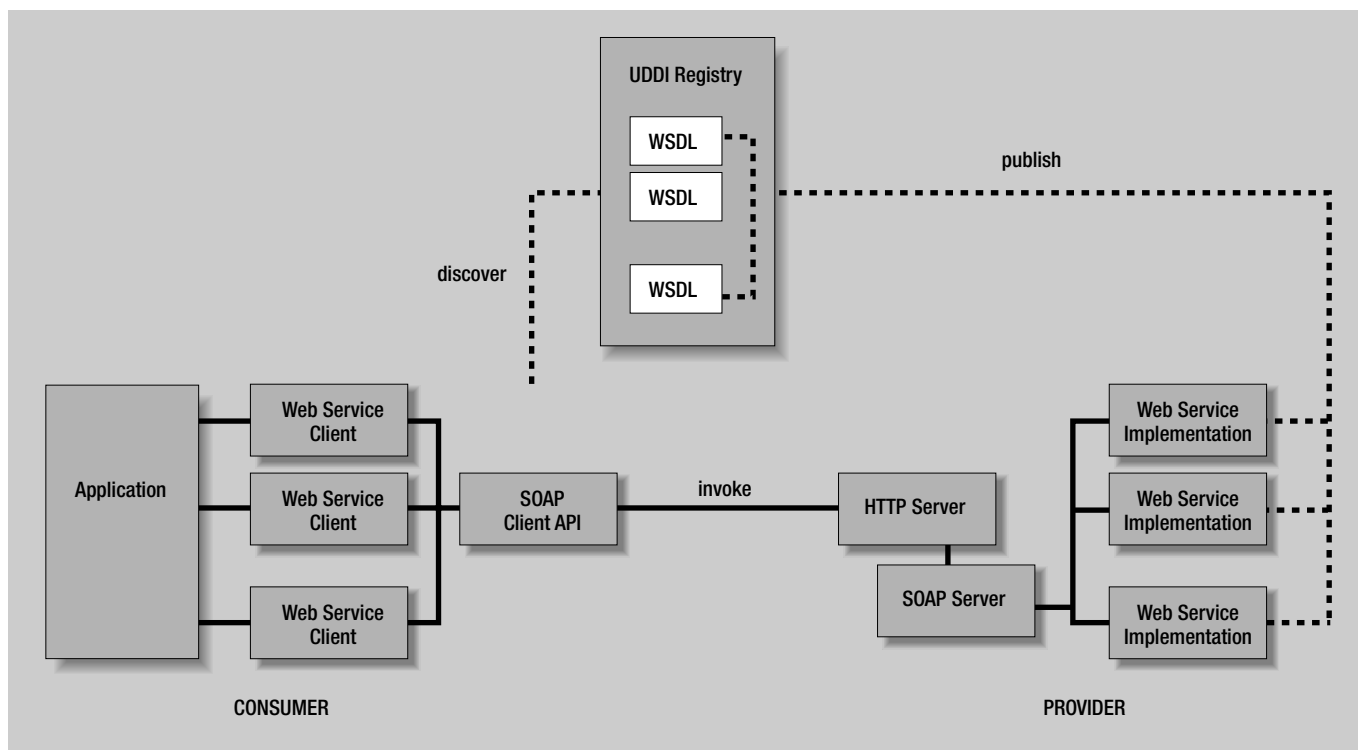
gebruikt als protocol voor de aanroep van een webservice. HTTP kent twee methoden: GET en POST. De eerste gebruiken we bij het benaderen van een normale webpagina: een URL (Uniform Resource Locator) wordt ingetypt en een HTML pagina wordt teruggestuurd. POST gebruiken we voor het opsturen van een document naar een bepaalde URL (bijvoorbeeld een file die we willen uploaden) waarna een document wordt teruggestuurd (bijvoorbeeld een pagina waarop bevestigd wordt dat de file succesvol geupload is). Voor webservices wordt gebruik gemaakt van HTTP POST en net als een normale webpagina bevindt een webservice zich op een bepaalde URL.

XML

De Extensible Markup Language is een inmiddels zeer breed geaccepteerde standaard voor het uitwisselen van gegevens. Bij webservices wordt een XML document via HTTP POST als aanroep naar de webservice gestuurd (gepost) en wordt (direct/synchroon) een XML document als antwoord teruggestuurd. Het XML document dat als aanroep gestuurd wordt bevat de parameters van de webservice in XML formaat en het teruggestuurde document bevat de return waarden in XML formaat.

SOAP

Het Simple Object Access Protocol is een standaard die oorspronkelijk is ontwikkeld door een groep bedrijven onder aanvoering van Microsoft en IBM, maar inmiddels is ondergebracht bij het World Wide Web Consortium (W3C). Deze organisatie



Afbeelding 1. Webservices op basis van SOAP, WSDL en UDDI

beheert onder andere ook de HTTP, XML en HTML standaards. SOAP is een standaard voor het in XML formaat specificeren van een Remote Procedure Call (RPC). Hierbij wordt in het XML document zowel aangegeven welke procedure (lees webservice) wordt aangeroepen als ook de parameters van de procedure. Het mooie van deze standaard is dat op basis van de SOAP API een webservice client gemaakt kan worden die het opbouwen, versturen en ontvangen van de XML documenten voor zijn rekening neemt. Vanuit de applicatie van de consumer kan deze webservice client aangeroepen worden als een normale procedure in de betreffende programmeertaal. Op deze manier is het heel eenvoudig om een webservice te gebruiken. Ook aan de kant van de provider maakt de SOAP standaard het leven van de programmeur gemakkelijk. De SOAP Server zet de SOAP aanroep automatisch om in een aanroep van de webservice implementation, de Java method of PL/SQL procedure of functie die de webservice implementeert. Bovendien wordt het resultaat van de webservice (d.w.z. de return waarden) automatisch in een XML document gezet en teruggestuurd.

WSDL

De Webservices Description Language is een op XML gebaseerde taal voor het beschrijven van webservices. Naast een XML Schema definitie van de parameters en return waarden bevat de WSDL beschrijving ook informatie betreffende de provider en de classificatie van de webservice.

UDDI

De Universal Description, Discovery, and Integration standaard beschrijft een repository, de UDDI Registry, van WSDL definities van webservices die op een standaardwijze doorzocht kan worden op zoek naar een webservice met een bepaalde functionaliteit.

In afbeelding 1 is aangegeven hoe het aanroepen, beschrijven en registreren van Webservices op basis van de SOAP, WSDL en UDDI standaards plaatsvindt. De provider heeft een aantal Webservices geïmplementeerd en beschreven in WSDL. De WSDL beschrijvingen zijn gepubliceerd in een UDDI Registry. De SOAP Server zorgt voor de afhandeling van de SOAP aanroepen die via de HTTP Server binnenkomen. De consumer "ontdekt" de webservice in de UDDI Registry. Op basis van de WSDL definitie van de webservice wordt een webservice Client ontwikkeld die met behulp van de SOAP Client API die webservice kan aanroepen. De applicatie roept de webservice client aan om de webservice te gebruiken.

Waarom een succes?

Waar is de verwachting dat webservices een succes zullen worden op gebaseerd? We hebben gezien dat de behoefte er is en dat de toepassingen er zijn. Daarnaast is een eerste belangrijke vaststelling dat webservices platform- en programmeer-

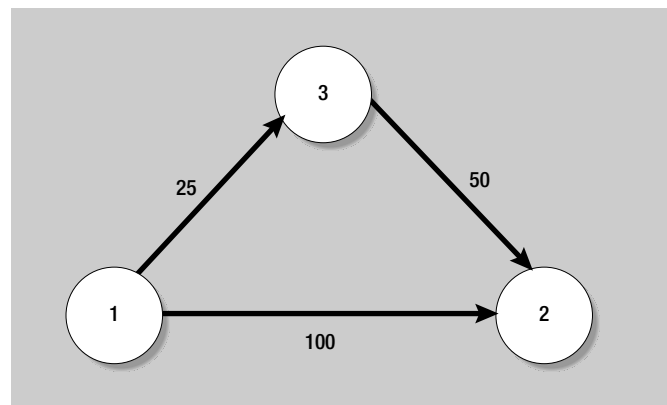
taal onafhankelijk zijn. Zowel Microsoft's .Net platform als het J2EE platform ondersteunen dezelfde, breed geaccepteerde en open, webservices standaards. Als tweede kunnen we vaststellen dat webservices eenvoudig zijn en dat de benodigde infrastructuur (netwerk- en applicatieserver) in de meeste gevallen al aanwezig is. Daarom is de verwachting dat webservices een succes zullen worden heel reëel.

Onze visie is dat het gebruik van webservices op korte termijn vooral tussen bekende business partners met lokale UDDI Registry's zal zijn en zich beperken tot het opvragen van informatie, eenvoudige B2B en A2A transacties (integratie) en content syndicatie. Later zullen er op basis van publieke UDDI Registry's webservices gebruikt gaan worden door nieuwe business partners en bij het beschikbaar komen van aanvullende standaards voor transactiemangement zal de toepassing van webservices zich uitbreiden tot complexe B2B en A2A transacties en business proces integratie. Tot besluit van dit artikel volgt nu een praktijkvoorbeeld, waarin een toepassing wordt beschreven die gebruik maakt van PL/SQL.

De ShortestPath Web Service

Webbservice in de praktijk: PL/SQL database package

De GRAPH database package hebben wij een paar jaar geleden ontwikkeld voor een cursus modulair programmeren in PL/SQL. De database package implementeert het uiterst efficiënte "Dijkstra's Algoritme" voor het bepalen van het kortste pad in een gerichte graaf. Een gerichte graaf is de wiskundige benaming van een netwerk van punten (nodes) en verbindingen (arcs) tussen deze punten. Bij een arc horen bepaalde "kosten" voor het volgen van de arc. Deze kosten betreffen bijvoorbeeld de afstand of reistijd tussen twee punten. Omdat het een gerichte graaf betreft hebben de arcs een bepaalde richting. Dit betekent dat indien een verbinding tussen twee punten in



Afbeelding 2. Een eenvoudige graaf.

beide richtingen gevolgd kan worden dit expliciet moet worden aangegeven door middel van twee arcs: één van A naar B en één van B naar A. Ook betekent dit dat de kosten (bijvoorbeeld de reistijd) om van A naar B te gaan kunnen verschillen van de kosten om van B naar A te gaan. Afbeelding 2 is een voorbeeld gegeven van een eenvoudige graaf.

De GRAPH database package is onafhankelijk van specifieke tabeldefinities. De graaf wordt geladen in een tweetal package variabelen die respectievelijk de nodes en de arcs bevatten, waarna meerdere kortste paden bepaald kunnen worden. Door deze onafhankelijkheid is de database package als module vanuit verschillende systemen te gebruiken. Dit maakt deze database package uitermate geschikt om als webservice beschikbaar te stellen. De specification van de GRAPH database package is als volgt.

```

Create or replace
Package graph
Is

Type nodes_type
Is
Table
of number(38, 0)
index
by binary_integer;

type arc_type
is
record
(
    from_node number(38, 0)
    ,
    to_node number(38, 0)
    ,
    cost number(28, 10)
);

type arcs_type
is
table
of arc_type
index
by binary_integer;

procedure load_graph
(
    p_nodes in nodes_type
    ,
    p_arcs in arcs_type
);

procedure get_shortest_path
(
    p_from_node in number
    ,
    p_to_node in number
    ,
    p_path out nodes_type
    ,
    p_cost out number
);

end;
```

Afbeelding 3a. Specification van de GRAPH database package.

In het onderstaande codevoorbeeld is te zien hoe de GRAPH database package gebruikt wordt om het kortste pad tussen punten 1 en 2 te bepalen in de eenvoudige graaf uit afbeelding 2.

```

Declare

l_nodes graph.nodes_type;
l_arcs graph.arcs_type;
l_path graph.nodes_type;
l_cost number;

begin

l_nodes(1) := 1;
l_nodes(2) := 2;
l_nodes(3) := 3;

l_arcs(1).from_node := 1;
l_arcs(1).to_node := 2;
l_arcs(1).cost := 100;

l_arcs(2).from_node := 1;
l_arcs(2).to_node := 3;
l_arcs(2).cost := 25;

l_arcs(3).from_node := 3;
l_arcs(3).to_node := 2;
l_arcs(3).cost := 50;

graph.load_graph(l_nodes, l_arcs);

graph.get_shortest_path(1, 2, l_path, l_cost);

dbms_output.put_line('shortest path =');
for i in 1..l_path.count loop
    dbms_output.put_line(l_path(i));
end loop;

dbms_output.put_line('cost = '||to_char(l_cost));

end;
```

Afbeelding 3b. Bepalen van het kortste pad tussen punten 1 en 2 in de graaf uit afbeelding 2.

Het resultaat van het uitvoeren van de code uit afbeelding 3 is als volgt:

```

Shortest path =
1
3
2
cost = 75
```

Afbeelding 4. Resultaat van het uitvoeren van de code uit afbeelding 3b.

Om de GRAPH database package als webservice beschikbaar te stellen hebben we een Java Servlet gemaakt die een XML document ontvangt met daarin de definitie van de graaf en de twee punten waartussen het kortste pad moet worden bepaald.

Het Servlet stuurt vervolgens een XML document terug met het kortste pad. Voor het beschikbaar stellen van deze webservice wordt hier niet gebruik gemaakt van SOAP.

```
<?xml version="1.0"?>
<getShortestPath>
  <graph>
    <nodes>
      <node id="1"/>
      <node id="2"/>
      <node id="3"/>
    </nodes>
    <arcs>
      <arc fromNode="1" toNode="2" cost="100"/>
      <arc fromNode="1" toNode="3" cost="25"/>
      <arc fromNode="3" toNode="2" cost="50"/>
    </arcs>
  </graph>
  <path fromNode="1" toNode="2"/>
</getShortestPath>
```

Afbeelding 5. Het XML document dat als aanroep naar de webservice wordt gestuurd.

```
<?xml version="1.0"?>
<shortestPath>
  <path>
    <node id="1"/>
    <node id="3"/>
    <node id="2"/>
  </path>
  <cost>75.0</cost>
</shortestPath>
```

Afbeelding 6. Het XML document dat als resultaat van de aanroep uit afbeelding 5 wordt teruggestuurd door de webservice.

De ABC Route Planner is een Internet applicatie waarmee de kortste of snelste route tussen grote steden in Noord-Amerika bepaald kan worden. Deze applicatie maakt gebruik van de onderstaande tabeldefinities.

ABC_CITIES		
Name	Null?	Type
ID	NOT NULL	NUMBER(38)
NAME	NOT NULL	VARCHAR2(60)

ABC_DISTANCES		
Name	Null?	Type
CIT_ID_FROM	NOT NULL	NUMBER(38)
CIT_ID_TO	NOT NULL	NUMBER(38)
DISTANCE	NOT NULL	NUMBER(5)
DRIVING_TIME	NOT NULL	NUMBER(57)

Afbeelding 7. Tabeldefinities van de ABC Route Planner

Tabel ABC_CITIES bevat de identificatie en naam van steden en tabel ABC_DISTANCES de afstand (in mijlen) en de reistijd in minuten tussen steden. De ABC Route Planner applicatie beschikt dus over de data (de graaf), maar implementeert niet zelf het algoritme om de kortste route te bepalen. Hiervoor zal gebruik worden gemaakt van de ShortestPath Web Service. Dit betekent dat de applicatie "onder de kap" een XML document opbouwt zoals dat in afbeelding 5, dit document verstuurt naar de webservice en op basis van het teruggegeven XML document de resultaten toont. In afbeeldingen 8 en 9 is te zien hoe de gebruiker een verzoek ingeeft en hoe de resultaten getoond worden.



Afbeelding 8. Het ingegeven verzoek tot het bepalen van de kortste weg tussen Montreal en Atlanta



Afbeelding 9. De bepaalde kortste route tussen Montreal en Atlanta

Erwin Groenendal

is technisch directeur van Cumquat Information Technology. Hij heeft 10 jaar ervaring met Oracle technologie. E-mail: erwin.groenendal@cumquat.nl. Cumquat richt zich op de toepassing van de nieuwste Oracle e-business technologie, XML en Java. Internet: www.cumquat.nl.