XML-WebServices

Steffen Forkmann

15. Juli 2004

Inhaltsverzeichnis

1 Was sind WebServices?

1.1 Einleitung

Ein XML-Webdienst ist eine Internetanwendung, die Daten und Dienste für andere Anwendungen (Clients) bereitstellt. Die Clients greifen dabei über vorhandene Webprotokolle und Datenformate, z.B. HTTP, XML und SOAP, auf die XML-Webdienste zu, unabhängig davon wie der betreffende XML-Webdienst implementiert ist. Sie sind wesentliches Element des Microsoft .NET-Programmiermodells.

> „Auf XML-Webdienste kann von jeder Sprache mit jedem Komponentenmodell und auf jedem Betriebssystem zugegriffen werden.“ [2]

XML-Webdienste verwenden meistens HTTP als zugrundeliegendes Transportprotokoll, dadurch wird die Übergabe von Funktionsanforderungen über Firmenfirewalls ohne weitere Konfigurationen ermöglicht. Durch das Formatieren der Eingabe- und Ausgabeparameter in XML ist die Anforderung nicht an eine bestimmte Komponententechnologie oder Objektaufrufkonvention gebunden. Des weiteren sind WebServices aber nicht an .NET beschränkt, sondern kommen zum Beispiel auch als Java-Webanwendungen vor. Die Idee dieser verteilten Anwendungen ist jedoch nicht neu. Vorläufer waren z.B. CORBA, DCOM und Java Beans (in J2EE). Durch die Festlegung von offenen Standards (SOAP, HTTP, XML) und der gezielten Ausrichtung von .NET auf WebServices wird die Entwicklung jedoch enorm erleichtert.

1.2 Einsatzmöglichkeiten

In Unternehmen tritt häufig das Problem auf, dass wertvolle Daten in isolierten Systemen schlummern, die nicht für den Datenaustausch mit anderen Systemen ausgerichtet sind. Wenn diese Daten jedoch für den Austausch mit anderen Systemen und Anwendungen nutzbar gemacht werden könnten, würde dies den Wert der Informationen erheblich steigern und gleichzeitig eine weitere Einnahmequelle für das Unternehmen bedeuten. Der Einsatz von WebServices wird demnach vor allem im Business-to-Business-Bereich stattfinden, wobei auch viele consumer-orientierte Anwendungen, wie im folgenden Abschnitt beschrieben, denkbar sind.

1.3 Beispiel Nachrichtenagentur

Ein typisches Einsatzgebiet eines WebService ist die Lieferung von Informationen aus einer Datenbank an verschiedene Client-Anwendungen. Ein Beispiel wäre die Lieferung von aktuellen Meldungen durch Nachrichtenagenturen (wie z.B. Reuters) an Medienunternehmen (Zeitungen, Fernsehen, Internetportale) und der damit verbundene Kostenausgleich. Die Realisierung eines solchen Projektes ist mit .NET WebServices nazu ein Kinderspiel, wenn man bedenkt, dass der Datenbankzugriff innerhalb weniger Augenblicke mit ADO.NET aufgebaut werden kann, die Bank-Überweisungen über WebService-Interfaces angeboten werden und die komplette Kommunikationssoftware von .NET-Klassen gekapselt wird.
In Abbildung 1 kann man ein solches Szenario betrachten.

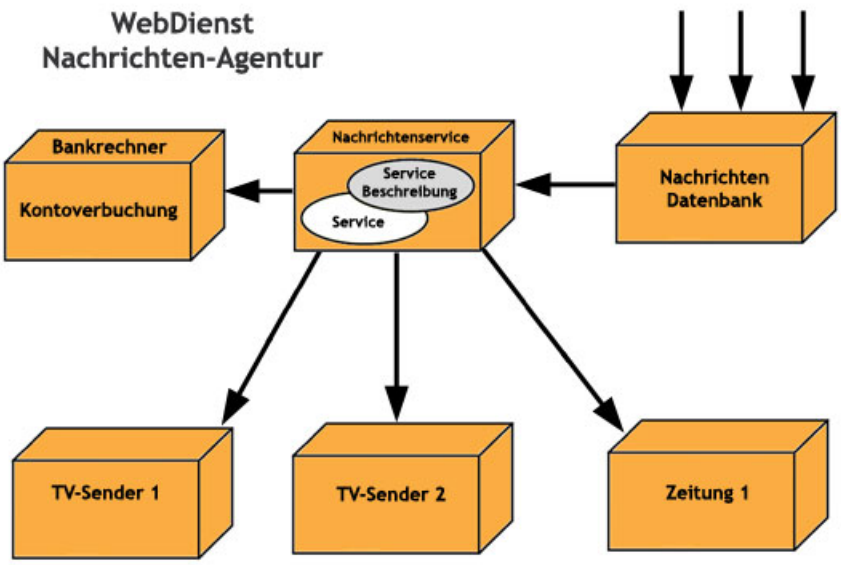

Abbildung 1: Einsatzbeispiel für WebServices

2 WebService-Architektur

2.1 Rollenverteilung

Die Architektur eines WebService besteht in idealisierter Form aus den folgenden drei Rollen (Three-Tier-Achitektur):

Dienstanbieter (Service Provider):

- Bietet einen Dienst (Funktionalität) über das Web an (SOAP)
- Beschreibt und annonciert den Dienst (WSDL)
- Übernimmt Implementierung, Betrieb und Wartung

Dienstanwender (Service Requestor):

- Nutzt angebotenen Dienst
- Sucht Dienste nach gewissen Kriterien (UDDI)
- Integriert Dienst in eigene Applikationen oder auch WebServices

Dienstmakler (Service Broker / Service Registry):

- Bietet Speicherung von Dienstbeschreibungen an
- Verwaltet kategorisierte Dienstbeschreibungen
- Ermöglicht automatisiertes Auffinden gespeicherter Dienste

2.2 Grundlegende Architektur

In der idealisierten WebService-Architektur könnten die Parteien wie in Abbildung 2 gezeigt zusammenarbeiten. Der Service Broker ist jedoch im Normalfall optional.

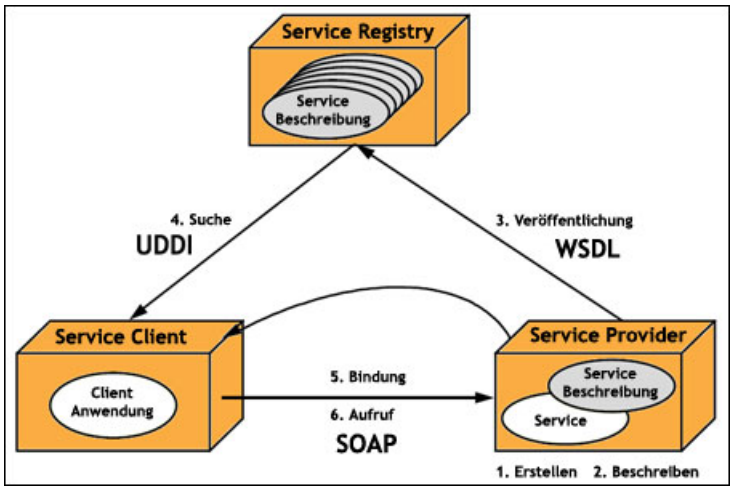

Abbildung 2: Grundlegende Architektur eines WebService

2.3 Architektur unter .NET

Wenn man die WebService-Archtitektur unter Micrsoft .NET betrachtet ergibt sich das in Abbildung 3 dargestellte Bild.

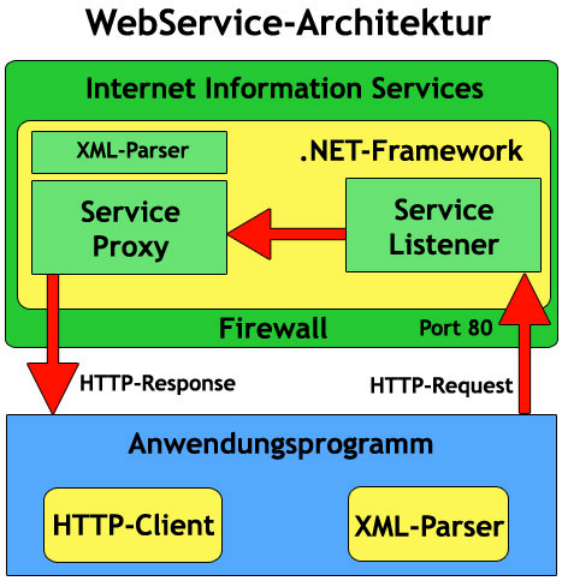

Abbildung 3: Grundlegende Architektur eines .NET-WebService

2.4 Datenaustausch

Der Datenaustausch wird meist mit der Hilfe von SOAP und HTTP durchgeführt. Alternativ zu SOAP kann der Datenaustausch auch über XML-RPC erfolgen.

2.4.1 SOAP

Das Simple Object Access Protocol SOAP ist ein plattformunabhängiges, XML-basiertes Protokoll, welches dazu dient, Anwendungen über das Web Datenobjekte austauschen zu lassen. Dabei ist es bemerkenswert, wie die einzelnen Spezifikationen kombiniert werden können. SOAP für sich gesehen ist unabhängig vom Transportprotokoll definiert, kann also neben der Standardvariante HTTP auch über FTP oder andere Protokolle arbeiten.
Im Kern legt SOAP fest, wie ein komplexes XML-Objekt in einen HTTP-"Umschlag" verpackt und versandt wird. Ein besonderer Vorteil: auf diese Art muss nicht für jeden möglichen Aufruf eines Programms von außen die Firewall des Rechners, auf dem es läuft, extra konfiguriert werden, weil SOAP-Datenaustausch als HTTP-Abfrage über einen standardmäßig freigegebenen Port erfolgt.

2.5 Auffinden des passenden WebService

Einen WebService kann man automatisch via UDDI finden. UDDI selbst basiert auf SOAP über HTTP. Weitere Protokolle sind angedacht und teilweise standardisiert, wie zum Beispiel WSIL.

2.5.1 UDDI

Das Universal Description, Discovery and Integration (UDDI)-Projekt ist ein Business-Verzeichnis. Die Initiative geht auf Microsoft und IBM zurück, besitzt aber mittlerweile viele weitere prominente Firmenmitglieder.
Ziel der UDDI-Datenbank ist es, das E-Business im Business-to-Business-Bereich voranzubringen. Dazu entsteht in einer XML-basierten Datenbank ein Verzeichnis von Unternehmen sowie ihren Produkten und Dienstleistungen. Die UDDI-Datenbank soll als Basis für die Integration von Angeboten in E-Business-Prozessen, beispielsweise in B2B-Marktplätzen dienen. UDDI definiert somit einen Standard, der die Suche nach Anbietern mit bestimmten technischen Standards oder Schnittstellen erleichtern soll.
Bislang werden von Ariba, Microsoft und IBM drei Datenbanken zu UDDI gehostet, Neueinträge und Updates werden untereinander synchronisiert. Die Eintragung ist kostenlos. Außer den Unternehmensdaten wird bei der Eintragung auch angegeben, welche Möglichkeiten und Präferenzen für E-Business-Kontakte bestehen. Auf diese Weise soll es möglich sein, mittels der UDDI-Datenbanken autmoatisiert Geschäftspartner zu finden, die sowohl vom Angebot als auch von den technischen Voraussetzung her geeignet sind.

2.6 Binden eines WebService

Für das Binden von WebServices dominiert der WSDL Standard. In WSDL kann man alle wichtigen Informationen ablegen, die man benötigt, um auf einen Web Service zuzugreifen. Ein Vorgänger von WSDL ist DISCO.

2.6.1 WSDL

Die WebServices Description Language (WSDL) definiert einen protokollunabhängigen XML Standard zur Beschreibung von WebDiensten bzw. zum Austausch von Nachrichten.
WSDL beschreibt einen Web Service durch 4 Komponenten:

1. Datentypen - Abstrakte Definition
2. Interface aller Funktionen - Abstrakte Definition
3. Binding (Transport Protokoll) - Konkrete Definition
4. Adresse des Services - Konkrete Definition

3 Ein einfaches Beispiel

Der Einsatz von WebServices gestaltet sich unter .NET sehr leicht. Der Entwickler kommt kaum in den Kontakt der dahinterliegenden Konzepte. WebDienste werden unter .NET immer von System.Web.Services abgeleitet und erben damit schon die gesamte Kommunkationsfunktionalität von der Basisklasse.

3.1 MyMath-WebDienst

Im folgenden werden die Möglichkeiten eines WebServices an einem ganz einfachen Beispiel demonstriert:

Listing 1: Ein einfacher Webdienst in C#

```
<%@ WebService Language="C#" Class="MyMath" %>
using System.Web.Services;

[WebService(
          Namespace="http://www.preisbooster.de",
          Name="Small Web Service in C#",
          Description="WebService with basic Math-functions.")]

public class MyMath
{
        [ WebMethod(Description="Adds two ints ....")]
        public int Add(int a, int b)
        {
                return a+b;
        }

        [ WebMethod(Description="Returns ....")]
        public int Sub(int a, int b)
        {
                return a-b;
        }
}
```

3.2 WebService einbinden

Um einen Client zu erzeugen, muss man sich als erstes die WSDL-Schnittstelle des WebDienstes besorgen. Mit Hilfe von Microsoft VisualStudio .NET ist dieser Vorgang innerhalb weniger Sekunden möglich, in dem man unter *Projekt/Webverweis hinzufügen* eine Web-Referenz auf den WebService erzeugt und dann den entsprechenden WebService in das Client-Projekt einbindet. Den Rest erledigt .NET beinahe zauberhaft selbstständig und der WebService ist im Clientprojekt ab diesem Zeitpunkt verfügbar:

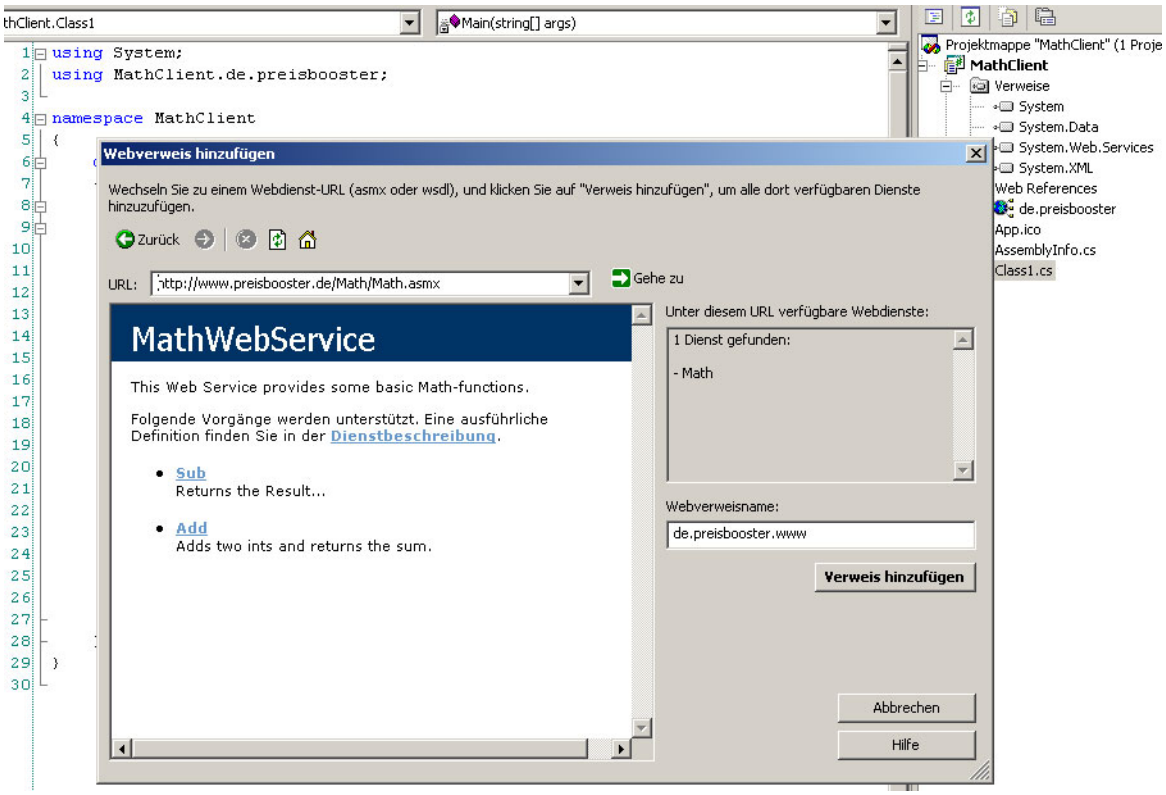

Abbildung 4: Verbindung mit WebDienst herstellen

3.3 MyMath-Client

Listing 2: Ein einfacher Konsolen-Client in C#

```
using System;
using MathClient.de.preisbooster;

namespace MathClient
{
        class Class1
        {
                [STAThread]
                static void Main(string[] args)
                {
                        try
                        {
                                MathWebService Service1
                                   = new MathWebService();
                                int c = Service1.Add(10,5);
                                Console.WriteLine("10 + 5 = "+c);
                                int d = Service1.Sub(10,5);
                                Console.WriteLine("10 - 5 = "+d);
                        }
                        catch
                        { Console.WriteLine("Konnte keine "
                           +"Verbindung zum WebService aufbauen.");
                  }

                        // Warten auf Enter
                        Console.ReadLine();
                }
        }
}
```

3.4 Fazit

Wenn man Informationen einem breiten Anwenderpool (Programmen) zur Verfügung stellen möchte, kommt man wohl um WebServices nicht mehr herum. Besonders die gute Implementierung in .NET in Zusammenarbeit mit dem mächtigen VisualStudio.NET sind sehr große Vorteile dieser Lösung. Allgemeinhin kann man sagen, dass durch die sogenannte „WebService-Allianz“ aus Microsoft, Sun und vielen anderen Firmen, WebServices immer mehr zur Standardlösung für B2B-Probleme werden. Positiv beurteilen kann man auch die klare Tendenz zur Standardisierung der Kommunikationsprotokolle durch das W3C, wenn auch einige Standards noch recht unsauber definiert sind. Die wirklichen Anwendungen sind jedoch (mit ein paar Ausnahmen wie z.B: Google, Amazon, ...) noch nicht erschienen.

Literatur

[1] Wolfgang Beer, Dietrich Birngruber, Hanspeter Mössenböck, and Albrecht Wöß. *Die .NET-Technologie*. dpunkt.verlag GmbH, 2003.

[2] Microsoft (Deutschland). Microsoft .net-installation. 2004.

[3] Klaus-Peter Fähnrich. *Vorlesung ComponentWare und WebServices*. Universität Leipzig, 2003.

[4] Mario Jeckle. Webservicearchitekturen w12. Tutorial, 2003.

[5] Andreas Kosch. *Crashkurs .NET für Delphianer*. Software & Support Verlag GmbH, 2003.

[6] Thomas Severins. Webservices. Tutorial, 2003.