Web Applications mit SOAP und RSS

Jonas Mitschang

Sommersemester 2005

Inhaltsverzeichnis

1 Einleitung **3**

2 WebServices **4**
- 2.1 Definition von WebService 4
- 2.2 Anwendungsgebiete 5
  - Enterprise Application Integration 5
  - Grid Computing 5
  - Business-to-Business 6
- 2.3 WebServices - Beispiel Fluggesellschaften 6
- 2.4 SOAP 6
  - Aufbau 7
  - Aussicht 7
- 2.5 UDDI 7
- 2.6 WSDL 8
- 2.7 SAML 8
  - Beispiel - Google API 8

3 RSS **11**
- 3.1 Versionen 11
- 3.2 Anwendung 11
- 3.3 Aufbau 12

4 Zusammenfassung **13**
- 4.1 Vorteile von WebServices mit SOAP 13
- 4.2 Nachteile von WebServices mit SOAP 13

1 Einleitung

Sie können sich zu jeder Zeit Webseiten ansehen und Informationen finden - kann Ihr Rechner das auch?

Heute lassen sich die meisten Informationen im Internet finden und nahezu beliebige Transaktionen und Buchungen durchführen. Diese Informationen sind meistens dem Benutzer vorbehalten und können auf Web Präsenzen eingesehen werden. Ich möchte mich im Rahmen dieser Ausarbeitung mit der Frage beschäftigen, wie der Rechner auf diese Resourcen zugreifen kann, ohne die Webinhalte umständlich parsen zu müssen.
Web Applications sind im Allgemeinen dynamische Erweiterungen des Webservers und lassen sich in zwei Bereiche gliedern:

- Darstellungsorientiert: herkömmliche Erweiterungen wie Foren, Boards, Shopsysteme, Newssysteme
- Dienstorientiert: WebServices

⇒„WebServices sind für den Rechner das, was für den Menschen die Webseite ist.“

2 WebServices

2.1 Definition von WebService

Ein WebService [2] ist eine modulare Software-Anwendung, die beispielsweise per XML [1] definiert und gefunden werden kann und die über einen eindeutigen Uniform Resource Identifier (URI) identifizierbar ist. WebServices dienen der direkten Interaktion zwischen Softwaresystemen unter Verwendung von XML formatierter Nachrichten.
Im Allgemeinen senden Clients Anfragen an einen WebService und dieser antwortet mit der gewünschten Information. WebServices sind nicht für menschliche Benutzer gedacht, sondern für Softwaresysteme, die automatisiert Daten austauschen und Funktionen auf entfernten Rechnern aufrufen.
WebServices orientieren sich an der Service Oriented Architecture (SOA) und bilden in diesem Kontext eine Funktionalität, die über eine standardisierte Schnittstelle in Anspruch genommen werden kann.
Komplexe Geschäftsprozesse lassen sich durch Aneinanderreihung von Service-Aufrufen realisieren. Die Programmlogik ist nicht in einem einzigen Programm zu finden, sondern ist über mehrere unabhängige Dienste verteilt, die auf ihren Aufgabenbereich zugeschnitten sind. Die SOA sieht eine Menge voneinander unabhängiger Dienste vor. Ein Dienst wird von einem Service Provider angeboten. Ein Servicekonsument stellt eine Anfrage (Service Request) an einen Dienst und bekommt daraufhin eine Antwort (Service Response) vom Anbieter. Die verfügbaren Dienste werden beim Servicebroker in einem Verzeichnis gespeichert.

Oft werden für WebServices als Basis SOAP, WSDL, UDDI und SAML eingesetzt, doch kann eine SOA prinzipiell auf jeder dienstbasierten Technologie aufgebaut werden. Im Folgenden soll tiefer auf die genannten Technologien, die jeweils auf XML basieren, eingegangen werden:

- SOAP - Das *Simple Object Access Protocol* dient der Kommunikation. Hiermit wird der eigentliche Aufruf gestartet.
- UDDI - *Universal Description Discovery and Integration* ist ein Verzeichnisdienst zur Registrierung von WebServices. Er ermöglicht das dynamische Finden des WebServices durch den Konsumenten.
- WSDL - Die *WebServices Description Language* beschreibt die unterstützten Methoden und deren Parameter eines Webservice.

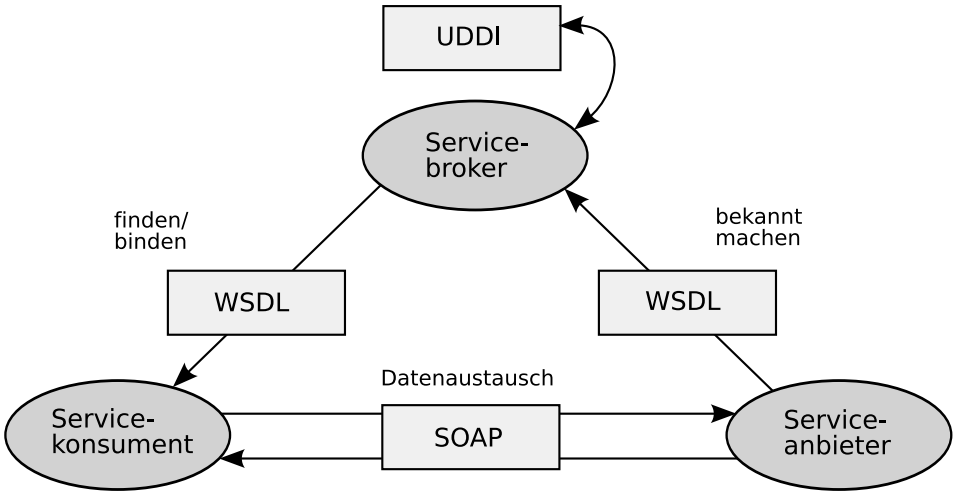

Abbildung 1: WebService Übersicht

- SAML - Mit der *Security Assertion Markup Language* werden die WebServices in ihren sonst nur oberflächlichen Sicherheitseigenschaften erweitert.

2.2 Anwendungsgebiete

WebServices stellen neue Ansätze für Enterprise Application Integration (EAI) und Grid-Computing dar. Allerdings liegt das geplante Haupteinsatzgebiet im Business-to-Business-Bereich (B2B).

Enterprise Application Integration

Mit der Enterprise Application Integration (EAI) wird versucht Geschäftsfunktionen, die über verschiedene Applikationen auf unterschiedlichen Plattformen verteilt sind, im Sinne der Geschäftsprozessintegration zu verbinden. Die EAI umfasst die Planung, Methoden und Software, um Anwendungssysteme zu integrieren, somit ist EAI die prozessorientierte Integration von Anwendungssystemen in heterogenen IT-Anwendungsarchitekturen.

Grid Computing

Grid-Computing bezeichnet unter anderem alle Methoden, die Rechenleistung vieler Computer innerhalb eines Netzwerks so zusammenzufassen, dass über den reinen Datenaustausch hinaus die (parallele) Lösung von rechenintensiven Problemen innerhalb einer Anwendungsdomäne ermöglicht wird. Die Kommunikation der Computer im Netzwerk

untereinander kann über WebServices und SOAP stattfinden.
Zum Beispiel arbeiten an dem Krebsforschungsprojekt auf der Grid.org Plattform [8] circa drei Millionen PCs, die erfolgversprechende Moleküle für Krebstherapeutika finden wollen.

Business-to-Business

Business-To-Business (auch B2B) steht für elektronische Kommunikationsbeziehungen zwischen verschiedenen Unternehmen. B2B stellt den Gegensatz zur Kommunikation mit Privatpersonen, Mitarbeitern oder der öffentlichen Verwaltung des Unternehmens dar.

Business-To-Business ist der älteste Teil des E-Business und der Hauptanwendungsbereich ist der elektronische Einkauf. B2B kann nicht nur mit Hilfe von WebServices realisiert werden, sondern auch mit Hilfe anderer Technologien wie zum Beispiel elektronische Marktplätze.

2.3 WebServices - Beispiel Fluggesellschaften

Die Interaktion zwischen Fluggesellschaften und Reisebüros läuft heute schon oft über WebServices: Die Reisebüros bieten auf ihrer Homepage Flüge verschiedener Fluggesellschaften an. Diese Flüge werden zur Laufzeit über UDDI von den Fluggesellschaften erfahren und in das Angebot übernommen.

Die Fluggesellschaften stellen Möglichkeiten zum Nachschlagen und sogar zum direkten Buchen von Flügen über deren WebServices bereit. Der Kunde kann nun bequem und vor allem zentral auf der Webpräsenz des Reisebüros die Preise und Termine der Flüge verschiedener Gesellschaften vergleichen.

2.4 SOAP

Das *Simple Object Access Protocoll*[3] wird in der Literatur oft synonym zu dem Begriff „WebService“ verwendet. Das liegt daran, dass die WebService Working Group des W3C SOAP als Bindung zum Zugriff auf WebServices definiert. SOAP ist dabei aber nicht die einzige Alternative, sondern lediglich diejenige, die als Standard definiert ist.

SOAP ist ein auf XML basierendes Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. Die gängigste Kombination ist SOAP über HTTP und TCP, aber es sind auch andere Transportschichten wie beispielsweise SMTP (Simple Mail Transfer Protocol: Email) oder FTP (File Transfer Protocol) definiert.

Aufbau

SOAP-Nachrichten sind nach dem Head-Body Pattern modelliert. Metainformationen werden im optionalen Head-Bereich (<soap:Header>) der Nachricht gespeichert. Diese Kopfdaten beinhalten Informationen über das Routing der Nachricht, eventuelle Verschlüsselung, Authentifizierungsinformationen oder über die Zugehörigkeit zu Transaktionen.

Die Nutzdaten der Nachricht, also die Aufruf- und Rückgabewerte, sind im Body (<soap:Body>) gespeichert und werden nur vom Empfänger interpretiert. Mögliche Zwischenstationen interpretieren nur die Kopfdaten.

Im Body kann sich ein optionales Fault-Element (<soap:Fault>) befinden, das auf Fehler wie ungültige Authentifizierung hinweist. Der Nachricht können je nach Transportprotokoll Anhänge folgen, die beispielsweise durch MIME-Mechanismen dargestellt sind.

Für eine Beipiel SOAP Nachricht siehe weiter unten das Google Beispiel.

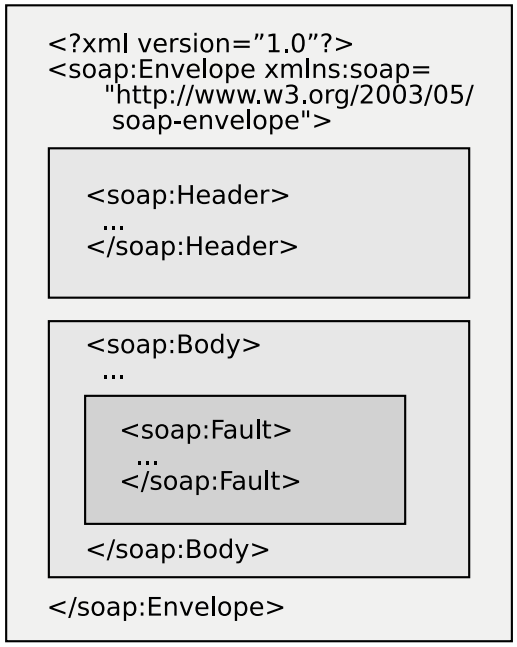

Abbildung 2: SOAP Nachricht

Aussicht

Früher war hauptsächlich HTTP als Transportschicht definiert aber heutige Implementationen abstrahieren vom Transport. Dadurch lässt sie sich fast beliebig austauschen. Neuere Erweiterungen betreffen den Bereich des Transports von binären Daten.

2.5 UDDI

UDDI [4] steht für *Universal Description Discovery and Integration* und ist eine Anwendung von SOAP. UDDI stellt mit Hilfe der SOAP-Schnittstelle einen Verzeichnisdienst bereit, der Unternehmen, ihre Daten und ihre Services enthält. UDDI lässt sich in drei Bereiche untergliedern:

- White Pages: Eine Art Telefonbuch, in der ein Namensregister eines Unternehmens mit entsprechenden Kontaktinformationen und Detailangaben aufgelistet ist.

- Yellow Pages: Bei den Yellow Pages handelt es sich um ein Branchenvereichnis, ähnlich den gelben Seiten. Es kann gemäß verschiedener Eigenschaften (wie z.B. Branche oder Ort) gesucht werden. Die Yellow Pages enthalten Verweise auf die White Pages, in denen dann wiederum Kontaktinformationen angefragt werden können, wenn ein Unternehmen gefunden wurde.
- Green Pages: In den Green Pages werden technische Informationen des Unternehmens gespeichert. Es können Daten wie angebotene WebServices, Geschäftsmodell und Geschäftsprozesse ermittelt werden.

2.6 WSDL

WSDL [5] ist eine Beschreibungssprache, mit der im XML-Schema WebServices definiert werden können. Es werden alle angebotenen Funktionen inklusive Parameter, Datentypen und Rückgabewerte beschrieben. Es ist auch möglich, komplexe Datenstrukturen und Arrays zu definieren. Es kann SOAP verwendet werden um WSDL Definitionen abzurufen.

Ein WSDL-Auszug folgt im Beispiel der Google API.

2.7 SAML

Da es für die Nutzung von WebServices keine geeigneten Sicherheitsmechanismen gab, entwickelte die Gruppe OASIS [6] im Jahr 2001 den SAML [7] Standard. Zum OASIS-Konsortium gehören Firmen wie Sun, SAP und IBM. SAML steht für Security Assertion Markup Language und beschreibt ein XML basierendes FrameWork zum Austausch von Sicherheitsinformationen, das auf folgende Fälle ausgelegt ist:

- Single Sign On: Man muss sich nur einmal anmelden und ist dann automatisch auch für weitere Anwendungen authorisiert.
- Verteilte Transaktionen: Sicherheitsinformationen werden zwischen mehreren Benutzern, die gemeinsam an einer Transaktion teilnehmen können, geteilt.
- Autorisierungsdienste: Es können dedizierte Maschinen existieren, die Berechtigungen für einen Dienst überprüfen.

Auf die sehr komplexe Sicherheitsbeschreibungssprache SAML kann ich aus Platzgründen an dieser Stelle nicht genauer eingehen.

Beispiel - Google API

Es gibt seit kurzem vom Betreiber der Suchmaschine Google WebServices [11], die die gleichen Möglichkeiten wie die Benutzerschnittstelle der Website anbieten. Es kann mit

SOAP Aufrufen nach Informationen im Internet gesucht werden oder Seiten in andere Sprachen übersetzt werden. Im Prinzip kann man auf alle Elemente der Suchmaschine zugreifen, unter anderem auch den Google-Cache.
Wenn man nicht auf WebServices zurückgreift muss man die Google Website parsen können. Das ist aber keine gleichwertige Alternative, weil man einen speziellen Parser für diese Seite implementieren müsste. Bei Änderungen an der Syntax müsste man in diesem Fall auch den Parser ändern.

WebService Aufruf

Ich habe die Google API mit den Programmiersprachen Java [9] und Perl [10] getestet. Der nötige Perl Code ist mit seinen sieben Zeilen im Vergleich zu den 74 Zeilen Java Code übersichtlicher:

Listing 1: Perl API

```
1 use SOAP::Lite;
2 $query='uni kaiserslautern';
3 $key='xxxxxxxxxxxxxxxxxxxxxxxx';
4 $search = SOAP::Lite->service("file:GoogleSearch.wsdl");
5 $result = $search->doGoogleSearch($key, $query, 1, 10,
6     "false", "", "false", "", "latin1", "latin1");
7 print "Results: $result->{'estimatedTotalResultsCount'}";
```

Den benötigten Key kann man bei Google kostenlos registrieren. Dieser Key berechtigt den Eigentümer zu 1000 Abfragen pro Tag.

WSDL-Definition

Die WSDL-Definition der Google API ist knapp 200 Zeilen groß, deshalb beschränke ich mich auf die im Perl Beispiel benutzte Definition von „doGoogleSearch“:

Listing 2: WSDL-Definition von doGoogleSearch

```
1 <message name="doGoogleSearch"
2     <part name="key"          type="xsd:string"/>
3     <part name="q"            type="xsd:string"/>
4     <part name="start"        type="xsd:int"/>
5     <part name="maxResults"   type="xsd:int"/>
6     ...
7 </message>
```

SOAP-Anfrage

Dies ist die SOAP-Anfrage, die der Client an den Server sendet (ohne HTTP-Header):

Listing 3: SOAP-Anfrage ohne HTTP-Header

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <soapenv:Envelope
3       xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
4       xmlns:xsd="http://www.w3.org/2001/XMLSchema"
5       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
6    <soapenv:Body>
7     <ns1:doGoogleSearch
8           soapenv:encodingStyle="http://schemas.xmlsoap.org/..."
9           xmlns:ns1="urn:CardValidator">
10      <key xsi:type="xsd:string">xxxxxxxxxxxxxxxxxxxxxxxx</number>
11      <q xsi:type="xsd:string">uni kaiserslautern</valid>
12      <start xsi:type="xsd:int">1</valid>
13      <maxResults xsi:type="xsd:int">10</valid>
14       ...
15    </ns1:doGoogleSearch>
16   </soapenv:Body>
17 </soapenv:Envelope>
```

3 RSS

RSS ist eine Familie von Dateiformaten, die fast alle auf XML basieren und in den verschiedenen Versionen auch verschieden benannt sind. Mit Hilfe von RSS können Nachrichten, Artikel und deren Kurzbeschreibungen gespeichert werden.

3.1 Versionen

- Rich Site Summary (RSS 0.9x) - Anfangs waren die Einträge limitiert, aber mit Version 0.92 wurde das Limit aufgehoben.
- RDF Site Summary (RSS 1.0) - RSS wurde um den RDF Standard erweitert, der Metainformationen für Einträge zulässt.
- Really Simple Syndication (RSS 2.0) - Die Version 2.0 setzt sich immer mehr als Standard durch. Sie erweitert RSS 0.9x (kein RDF mehr).
- Really Simple Syndication (RSS 3.0) - RSS 3.0 ist kaum verbreitet und benutzt als einziger Standard kein XML.

3.2 Anwendung

Diese RSS-Dateien werden RSS-Feeds oder Newsfeeds genannt und bestehen aus einer XML-Datei, auf die über eine eindeutige URL zugegriffen werden kann. Die XML-Datei enthält kein Layout und keine Formatierungen sondern lediglich die Informationen, die dann auf der Webpräsenz mit dem Design der Webseite versehen werden können.

Ursprünglich wurden RSS-Feeds in Online-Tagebüchern (Weblogs oder Blogs, Kunstwort aus Web und Log) verwendet aber inzwischen setzen die meisten Nachrichtenseiten, die ihren Inhalt häufig ändern müssen (z.b. Heise [12] oder Spiegel [13]) RSS-Feeds ein.

Das Bedürfnis, Erlebnisse und Erkenntnisse online der Öffentlichkeit zugänglich zu machen, scheint unter den Internet-Nutzern rapide zu wachsen. Momentan gibt es weltweit etwa 70 Millionen Blogs. Die Zahl der Blogs ist in Deutschland im vergangenen Jahr geradezu explodiert: Während im Juli 2004 noch 7500 deutsche Blogs online waren, ist die Zahl nun auf über 280.000 angestiegen. Man zählt RSS zu den ersten Anwendungsgebieten des Semantic Web.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rss version="2.0">
 <channel>
  <title>FS-Info Blog RSS</title>
  <description>RSS-Feed des Blogs der FS Informatik</description>
  <lastBuildDate>Fri, 03 Jun 2005 12:26:16 +0100</lastBuildDate>
  <copyright>Copyright 2005, Fachschaft Informatik, TU Kl</copyright>
  <webMaster>webmaster@fachschaft.informatik.uni-kl.de</webMaster>
  <docs>http://blogs.law.harvard.edu/tech/rss</docs>

  <item>
   <title>Hitchhiker's Guide to the University - Update</title>
   <link>http://www.fachschaft.informatik.uni-kl.de/blog/?eid=31</link>
   <description>Das Einfuehrungsbuechlein der Fachschaft ... </description>
   <pubDate>Fri, 15 Apr 2005 08:20:41 +0100</pubDate>
  </item>

  ...
 </channel>
</rss>
```

Abbildung 1: Struktur eines RSS Feed

3.3 Aufbau

RSS Dateien haben einen Header, in dem Eigenschaften wie Titel und Beschreibung des RSS Feed angegeben werden können. Die eigentlichen Nachrichten befinden sich jeweils in „item“-Elementen und haben zusätzliche Attribute wie Kurzfassung der Nachricht oder einen Link auf eine ausführliche Beschreibung.

Dies ist ein beispielhafter Ausschnitt aus dem RSS Feed der Fachschaft Informatik [14] der Technischen Universität Kaiserslautern.

4 Zusammenfassung

4.1 Vorteile von WebServices mit SOAP

- Durch die verwendeten offenen Standards werden Lizenzkosten vermieden.
- Es treten weniger Probleme mit Firewalls auf, weil das verwendete HTTP-Protokoll (auch FTP, SMTP) zur Datenübertragung meist ohnehin freigeschaltet ist. Bei vergleichbaren Technologien wie CORBA, DCOM oder auch Java RMI ist eine Umstellung der Firewall nötig.
- Bei Entwicklungen, die auf XML aufbauen, ist prinzipiell eine gute Tool-Unterstützung gegeben.
- Die Barrieren zum Einstieg sind vergleichsweise niedrig.

4.2 Nachteile von WebServices mit SOAP

- Bei Verwendung von XML-Protokollen entsteht immer ein deutlicher Overhead zu den Nutzdaten. Im Fall von SOAP nehmen die Daten bei einem Request im Schnitt um den Faktor 25 zu. Bei der Antwort des Servers (bei der es sich ja auch nur im einen einzigen Integer handeln kann) kann ein Overhead von 500% und mehr entstehen kann.
- Die Hauptschwierigkeit bei der Umsetzung von WebServices dürften Sicherheitsaspekte betreffen. So ist beim Transport zu beachten, dass wichtige WebServices verschlüsselt werden oder eine Authentifizierung stattfinden kann. Dabei muss abgewägt werden, welche Technologie (HTTPS, XML-Signature, XML-Encryption, SAML) eingesetzt werden soll.
- Es ist mehr Know How erforderlich, als z.B. für Remote Method Invocation von Java. Programmiersprachen, mit denen man WebServices einbinden will, brauchen spezielle Libraries (z.B DOM).

Literaturverzeichnis

[1] W3C XML - http://www.w3.org/XML/

[2] W3C WebServices - http://www.w3.org/2002/ws

[3] W3C SOAP - http://www.w3.org/TR/SOAP

[4] UDDI - http://www.uddi.org

[5] W3C WSDL - http://www.w3.org/TR/wsdl

[6] OASIS - http://www.oasis-open.org

[7] OASIS SAML - http://www.oasis-open.org/committees/security/#documents

[8] Grid.org - http://www.grid.org

[9] Sun Microsystems Java - http://java.sun.com

[10] Practical Extraction and Report Language - http://www.perl.com

[11] Google WebServices - http://www.google.com/apis

[12] Heise - http://www.heise.de

[13] Spiegel Online - http://www.spiegel.de

[14] RSS Feed der Fachschaft Informatik
http://www.fachschaft.informatik.uni-kl.de/blog/rss20.xml