

Web Services: Modellierung externer Abläufe

Alexander Elsas

*Institut für Wirtschaftsinformatik
Johann Wolfgang Goethe-Universität Frankfurt am Main
D-60054 Frankfurt am Main, Mertonstraße 17
elsas@wivi.uni-frankfurt.de*

Schlagnworte: Web Services, SOAP, WSDL, UDDI, Ablaufmodellierung, WSFL, XLANG, BPEL4WS, BPML

Abstract: Web Services etablieren sich als Standard für verteilte Applikationen. Aufbauend auf den Kernprotokollen SOAP, WSDL und UDDI sind dabei Ablaufmodellierungssprachen in der Entwicklung. Dieser Beitrag stellt die Grundlagen des Web Services-Framework, die darauf aufbauenden verschiedenen Ablaufmodellierungssprachen und die Bedeutung von übergreifenden Standards zur Ablaufmodellierung dar.

1. Web Services

1.1. Konzept

Hauptziel des Web Service-Konzepts ist die automatisierte Kommunikation bzw. Interaktion von verschiedenen, verteilten Anwendungen.

Das *World Wide Web Consortium* (W3C) definiert einen Web Service als eine Softwareanwendung „whose interfaces and bindings are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet-based protocols.“¹

Im Gegensatz zu traditionellen IT-Systemen, die man als eng gekoppelte Systeme charakterisieren kann, implementieren Web Services ein lose gekoppeltes System.

Das Web Service-Framework besteht aus drei grundlegenden Diensten, die eine sog. *Service-orientierte Architektur* (SOA) darstellen und mit Hilfe von in XML definierten Protokollen realisiert werden:

¹ *Austin, Daniel; Barbir, Abbie; Garg, Sharad* (Hrsg.): *Web Services Architecture Requirements – W3C Working Draft 29 April 2002*; URL: <http://www.w3.org/TR/2002/WD-wsa-reqs-20020429> (6.8.2002).

- **Kommunikation:** nachrichtenorientierte Kommunikation zwischen Applikationen, realisiert durch das *Service Oriented Architecture Protocol* (SOAP);
- **Dienstbeschreibung:** Definition und Beschreibung eines Dienstangebots, realisiert durch die *Web Services Description Language* (WSDL);
- **Dienstentdeckung:** Publizieren und Auffinden eines Dienstes, realisiert durch *Universal Description, Discovery, and Integration* (UDDI).

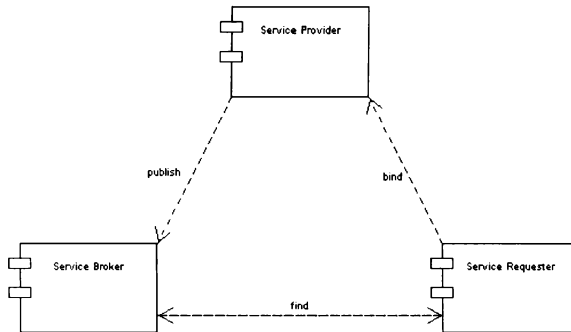


Abbildung 1: Service-orientierte Architektur

Dieses Zusammenspiel wird in Abbildung 1 veranschaulicht:²

- Ein bestimmter Web Service übermittelt seine WSDL-Beschreibung an ein UDDI-Verzeichnis (publish).
- Ein Client durchsucht das Verzeichnis nach einer bestimmten Dienstdefinition (find).
- Client und Service kommunizieren dann direkt mit Hilfe von SOAP auf Grundlage der im Verzeichnis hinterlegten WSDL-Spezifikationen des Dienstes (bind).

1.2. Kernprotokolle

1.2.1. SOAP

Das *Service Oriented Architecture Protocol* (SOAP)³ ist ursprünglich (noch als *Simple Object Access Protocol* bezeichnet) eine gemeinsame

² Vgl. IBM (Hrsg.): *Web Services for Remote Portals (WSRP)*; URL: <http://www-106.ibm.com/developerworks/library/ws-wsrp/> (6.1.2003).

³ Vgl. *Mitra, Nilo* (Hrsg.): *SOAP Version 1.2 Part 0: Primer*; 2002, URL: <http://www.w3.org/TR/soap12-part0/> (26.8.2002).

Entwicklung von Microsoft, IBM und einigen anderen Firmen, dessen Weiterentwicklung mittlerweile in den Händen des W3C liegt.⁴

SOAP ist ein XML-basiertes Protokoll zum Austausch von Nachrichten und für sog. *Remote Procedure Calls* (RPCs), dem Aufruf von Programmen bzw. Programmfragmenten auf entfernten Rechnern. Es definiert kein neues Transportprotokoll, sondern baut auf den vorhandenen Internet-Transportprotokollen wie HTTP und SMTP auf. Somit ist SOAP die Verbindung des Web Service-Konzepts zur bestehenden Infrastruktur des Internets.

Die Kommunikation zwischen Applikationen mit SOAP ist gekennzeichnet durch:⁵

- Die Kommunikation erfolgt nachrichtenorientiert, d.h. durch den Austausch von Textnachrichten, deren Format innerhalb von SOAP definiert ist.
- Eine SOAP-Nachricht ist ein XML-Element mit zwei Kind-Elementen: header und body.
- SOAP-Nachrichten enthalten Informationen für den Empfänger, wie sie verarbeitet werden sollen.
- Zusätzlich können Informationen, wer eine Nachricht verarbeiten soll, in einer SOAP-Nachricht enthalten sein.
- Mit Hilfe des SOAP-Protokolls können auch RPC realisiert werden.

Abbildung 2 zeigt eine typische, vereinfachte SOAP-Nachricht, die ein elektronisches Flugticket darstellt.

```

POST /travelservice
SOAPAction: "http://www.cybertravel.com/checkin"
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP: Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Body>
    <et:eTicket xmlns:et="http://www.cybertravel.com/eticket/schema">
      <et:passengerName first="Alexander" last="Elsas"/>
      <et:flightInfo
        segment="FRA-SZG"
        airline="LH"
        class="C"
        flight="2489"
        departureDate="2003-02-19"
        departureTime="1905"
        arrivalDate="2003-02-11"
        arrivalTime="2030" />
      </et:eTicket>
    </SOAP:Body>
  </SOAP:Envelope>

```

Abbildung 2: Elektronisches Flugticket als SOAP-Nachricht

⁴ Vgl. *Box, Don: A Brief History of SOAP; 2001; URL: <http://www.xml.com/pub/a/2001/04/04/soap.html> (6.8.2002).*

⁵ Vgl. *Curbera, Francisco; Dufiler, Matthew; Khalaf, Rania; Nagy, William; Mukhi, Nirmal, Weerawarana, Sanjiva: Unraveling the Web Services Web – An Introduction to SOAP, WSDL, and UDDI; in: IEEE Internet Computing, March / April 2002, S. 86–93.*

1.2.2. WSDL

Die *Web Services Description Language* (WSDL) beschreibt Web Services als eine Kollektion von Kommunikationsendpunkten, die Nachrichten austauschen. Eine komplette Beschreibung eines Web Services besteht aus zwei Teilen, dem *abstrakten Interface* und den *Protokoll-spezifischen Details*.⁶

- Das **abstrakte Interface** ist eine Dienstbeschreibung auf Anwendungsebene. Ein Interface besteht dabei aus drei Hauptkomponenten: *Vokabular*, *Nachrichten* und *Interaktionen*.
 - Vokabular: Die meisten Web Services verwenden die Datentypen der XML Schema Definition (XSD)⁷ als Standard.
 - Nachrichten: Sie werden aus Elementen des Vokabulars gebildet. Zusammengehörige Eingangs- und Ausgangsnachrichten sind eine *Operation*, die Menge der erlaubten Operationen an einem Endpunkt bildet einen *portType*.
 - Interaktionen: Werden durch Operationen und portTypes definiert.
- Die **Protokoll-spezifischen Details** müssen befolgt werden, um Zugriff auf von den Endpunkten angebotene Web Services zu erlangen. Diese sog. *Bindungen* (bindings) beschreiben, welches Kommunikationsprotokoll benutzt werden muss, wie die individuelle Dienstinteraktion über das Protokoll erreicht wird und wo (an welcher Netzwerk-Adresse) die Kommunikation endet.

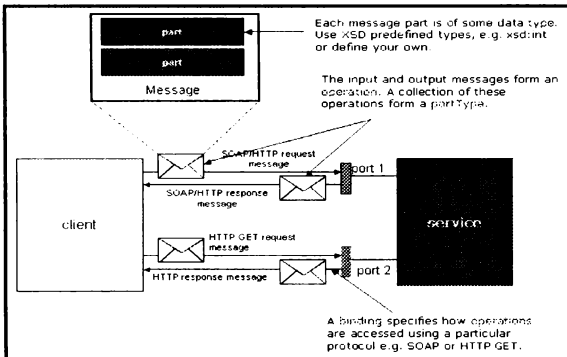


Abbildung 3: Aufruf eines Web Service (Quelle Shohoud)

⁶ Vgl. *Weerawarana, Sanjiva; Chinnici, Roberto; Gudgin, Martin; Moreau, Jean-Jacques* (Hrsg.): *Web Services Description Language (WSDL) Version 1.2 – W3C Working Draft 9 July 2002*; URL: <http://www.w3.org/TR/wsd112> (6.8.2002).

⁷ Vgl. *W3schools* (Hrsg.): *XML Schema Tutorial*; URL: <http://www.w3schools.com/schema> (14.4.2003).

Abbildung 3 zeigt den Zugriff auf einen Web Service und die entsprechenden WSDL-Begriffe. Dieser Zugriff kann einmal über SOAP/HTTP erfolgen, zum anderen auf dem traditionellen Weg über HTTP. Jeder Aufruf besteht dabei aus einer Anfragenachricht (request message) und einer Antwortnachricht (response message).⁸

1.2.3. UDDI

Mit Hilfe von *Universal Description, Discovery, and Integration* (UDDI) werden Online-Verzeichnisse von Web Services realisiert. Ein UDDI-Verzeichnis enthält dabei zwei grundlegende Spezifikationen:⁹

- **Verzeichnisstruktur:** welche Informationen über einen Web Service verzeichnet werden und wie diese zu codieren sind.
- **Verzeichnisoperation:** stellt ein sog. Application Programming Interface (API) dar, welches Abfragen und Veränderungen an den Verzeichniseinträgen erlaubt.

Ein UDDI-Verzeichnis beinhaltet drei verschiedene Arten von Informationen über einen Web Service, diese werden analog zu Telefonbüchern bezeichnet:

- Namen und Ansprechpartner (Weiße Seiten, white pages),
- Eine Kategorisierung von Geschäfts- und Dienstypen (Gelbe Seiten, yellow pages),
- Technische Details (Grüne Seiten, green pages).

2. Ablaufmodellierung

Zur Realisierung komplexer, verteilter Applikationen ist insbesondere die Modellierung der zugrunde liegenden Abläufe von Bedeutung. Diese externen Abläufe sind dabei durchaus über den Blickwinkel des E-Government hinausgehend.

So stellt der Vorgang *Anmeldung eines Kfz* zunächst einen Government-to-Citizen-Vorgang dar, der unter einer entsprechenden Lebenslage Eingang in z.B. ein One-Stop-Government-Portal finden kann.

Zur kompletten Umsetzung dieses Ablaufs ist aber die Integration eines Schildermachers wünschenswert, dies sprengt als Business-to-Consumer-Transaktion den E-Government-Blickwinkel. Abbildung 4 veranschaulicht diesen Fall.

⁸ Vgl. *Shohoud, Yasser*: Introduction to WSDL; URL: <http://www.learnxmlws.com/tutors/wsdl/wsdl.aspx> (7.8.2002).

⁹ Vgl. UDDI.org (Hrsg.): UDDI Technical White Paper; 2000, URL: http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf (7.8.2002).

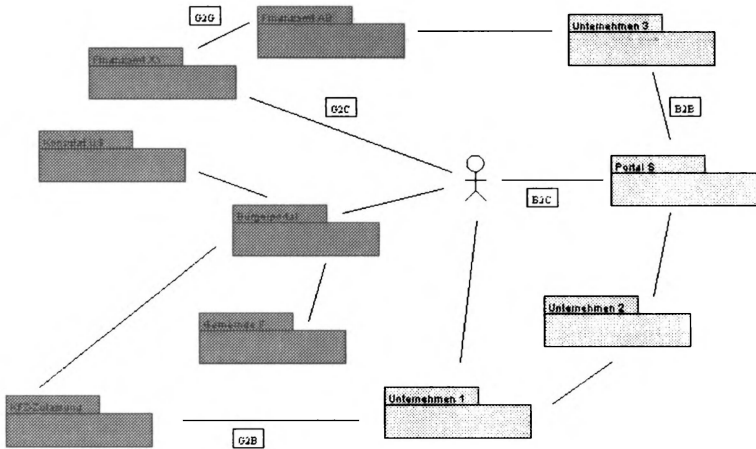


Abbildung 4: Übergreifende Transaktionen

Wünschenswert sind somit Ablaufmodellierungssprachen, die Web Services unterstützen und nicht E-Government-spezifisch sind, sondern anwendungskontextunabhängige Modellierung und Implementierung erlauben.

Einen Überblick über die im Kontext von Web Services entwickelten Ablaufmodellierungssprachen gibt die folgende Tabelle 1.

Akronym	Bezeichnung	Entwickler	URL
WSFL	Web Services Flow Language	IBM	http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
XLANG		Microsoft	http://www.gotdotnet.com/team/xml_ws_specs/xlang-c/default.htm
BPEL4WS	Business Process Execution Language for Web Services	IBM, Microsoft, BEA	http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/
BPML	Business Process Modeling Language	BPML.org	http://www.bpml.org/bpml-spec.esp

Tabelle 1: Ablaufmodellierungssprachen im Web Services-Framework

Dabei ist eine Konvergenz in Richtung auf BPML festzustellen. Die zunächst konkurrierenden Ansätze XLANG von Microsoft und WSFL von IBM münden in den gemeinsamen Ansatz BPEL4WS, dieser wieder-

um wird wohl in BPML aufgehen. Hier ist also noch kein gefestigter, etablierter Standard entstanden.

3. Fazit

Als Standard zur Implementierung von verteilten Applikationen etabliert sich das Web Service-Konzept, das insbesondere Unterstützung von führenden IT-Anbietern erhält.

So sind das .NET-Framework von Microsoft und J2EE von Sun die prominenten Beispiele dafür. Die Kernprotokolle des Web Service-Konzepts SOAP, WSDL und UDDI sind dabei schon akzeptiert und etabliert. Weitergehende Ansätze wie z. B. Ablaufmodellierungssprachen befinden sich noch in der Entwicklung, hier ist noch kein gefestigter Entwicklungsstand erreicht.

Die breite Industrieunterstützung und auch der Aspekt der übergreifenden Kommunikation zwischen E-Government und E-Commerce-Applikationen legt es nahe, den Web Service-Ansatz als technische Infrastruktur für E-Government-Anwendungen zu wählen.