
Aus- und Rückblick

Heutige Vorlesung

Web Services in der Praxis

- Wie wird SOAP/WSDL verwendet?
- Anforderungen der Praxis
- Erweiterungen von SOAP/WSDL

Semantic Web

- Vergleich mit XML

Rückblick

- Was waren die wichtigsten Lernziele?

Web Services in der Praxis

Web Services in der Praxis

Wie wird SOAP/WSDL verwendet?

- Web Services austesten, aufrufen und anbieten
- Dynamische vs. statische Einbindung

Anforderungen der Praxis

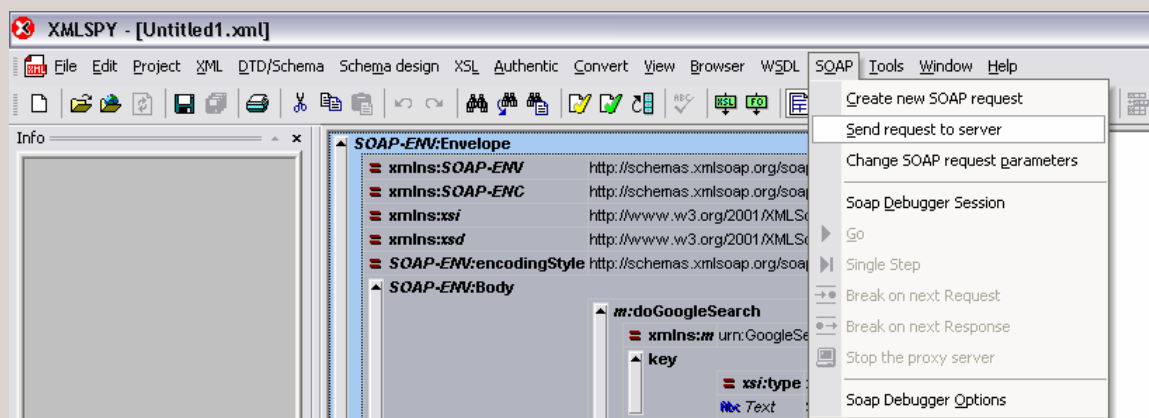
- Web Services im E-Business

Erweiterungen von SOAP/WSDL

- WS Reliable Messaging, WS Addressing, WS Security, WS Policy, BPEL



- externen Web Service austesten:
Funktionalität, Reaktionszeit
- eigenen Web Service austesten:
Testen, ob eigener Web-Dienst auch über WSDL-
Beschreibung abrufbar



WSDL → SOAP-Anfrage → SOAP-Antwort

- per Knopfdruck: WSDL-Beschreibung → SOAP-Anfrage
- Parameter der SOAP-Anfrage editieren
- per Knopfdruck: SOAP-Anfrage an Web Service senden
- Antwort als SOAP-Nachricht sichtbar



WSDL → Stubs

- Aus WSDL-Beschreibung *automatisch* Schnittstelle zur Anwendung (Stubs) generieren
- Stubs abstrahieren von SOAP und vom konkreten Übertragungsprotokoll.
- Web Service erscheint als *lokale* Bibliothek.



Anwendungsprogramm → Web Service, einschl. WSDL

- Anwendungslogik implementieren
- per Knopfdruck: Anwendung → Web Service, einschließlich WSDL-Beschreibung
- Übersetzung
Datenstruktur → Datentypen eines XML-Schemas
nicht immer automatisch möglich

Welches Werkzeug verwenden?



XMLSpy

- + schnelles austesten eines Web Service
- + SOAP-Anfrage und -Antwort sichtbar
- keine Anwendungsentwicklung möglich



.net

- + Anwendungsentwicklung möglich
- aufwendige Installation



Apache axis

- + Anwendungsentwicklung unter Java möglich
- aufwendige Installation



Die Vision: dynamisch Einbindung

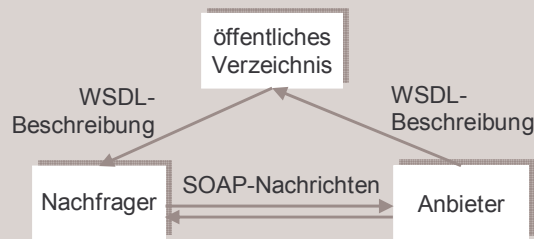


Beispiel

- personalisierte Informations-Webseite
- Benutzer möchte Echtzeitkurse von bestimmter Aktie einer bestimmten Börse
- Informations-Webseite sucht in öffentlichen Verzeichnis passenden Web Service für Echtzeitkurse
- Informations-Webseite bindet den Web Service automatisch ein



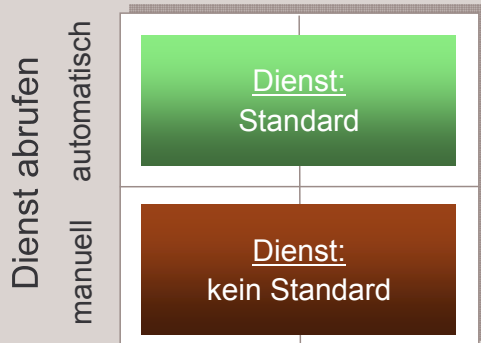
⇒ dynamische Einbindung zur Laufzeit



1. Anwendung sucht in Verzeichnis passenden Web Service
2. Suchergebnis: WSDL-Beschreibung
3. WSDL-Beschreibung → Stubs
4. Anwendung ruft mit Hilfe der Stubs Web Service auf

⇒ Dienst muss automatisch gefunden und aufgerufen werden.

Schritt 4: Aufruf des Dienstes mit Stubs



- gegeben: WSDL-Beschreibung
- Um Web Service *automatisch* mit Stubs aufzurufen, muss er **standardisiert** sein:
- WSDL beschreibt zwar *Syntax* der Schnittstelle, nicht aber die *Bedeutung* der Prozedur/Parameter.
- Bedeutung muss außerhalb von WSDL festgelegt (standardisiert) werden

Beispiel



Float Aktienkurs(Integer WKN, String Boersenplatz)

- Stub kann automatisch aus WSDL-Beschreibung generiert werden

aber:

- Was bedeutet WKN?
- Was bedeutet Boersenplatz?
Wie wird z.B. der Xetra-Handel in Frankfurt kodiert?
- Was bedeutet Aktienkurs?
Echtzeit oder verzögert? Wie lange verzögert?
- Was bedeutet Ergebnistyp Float?
Welche Währung?

Beispiel



Float Aktienkurs(Integer WKN, String Boersenplatz)

- solcher Stub kann automatisch aus WSDL-Beschreibung generiert werden

aber:

- Was bedeutet WKN?
- Was bedeutet Boersenplatz?
Wie wird der xetra-H... in Frankfurt kodiert?
- Was bedeutet ...
Echtzeit o... ert? Wie lange verzögert?
- Was bedeu... Ergebnistyp Float?
Welche Währung?

muss außerhalb von WSDL festgelegt werden!

automatischer Aufruf zur Laufzeit

- Schritt 3: Übersetzung WSDL → Stubs erfolgt automatisch
- Schritt 4: Integration von Stubs in Anwendungslogik erfolgt automatisch

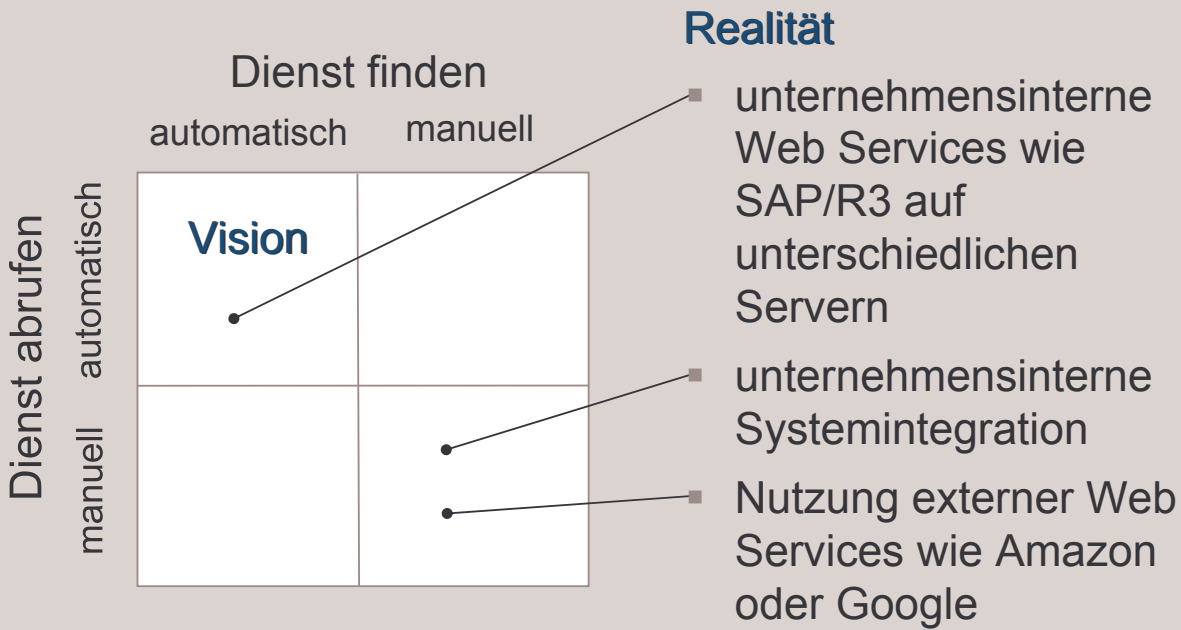
manueller Aufruf zur Entwicklungszeit

- Schritt 3: Übersetzung WSDL → Stubs erfolgt automatisch
- Schritt 4: Integration von Stubs in Anwendungslogik programmiert Entwickler

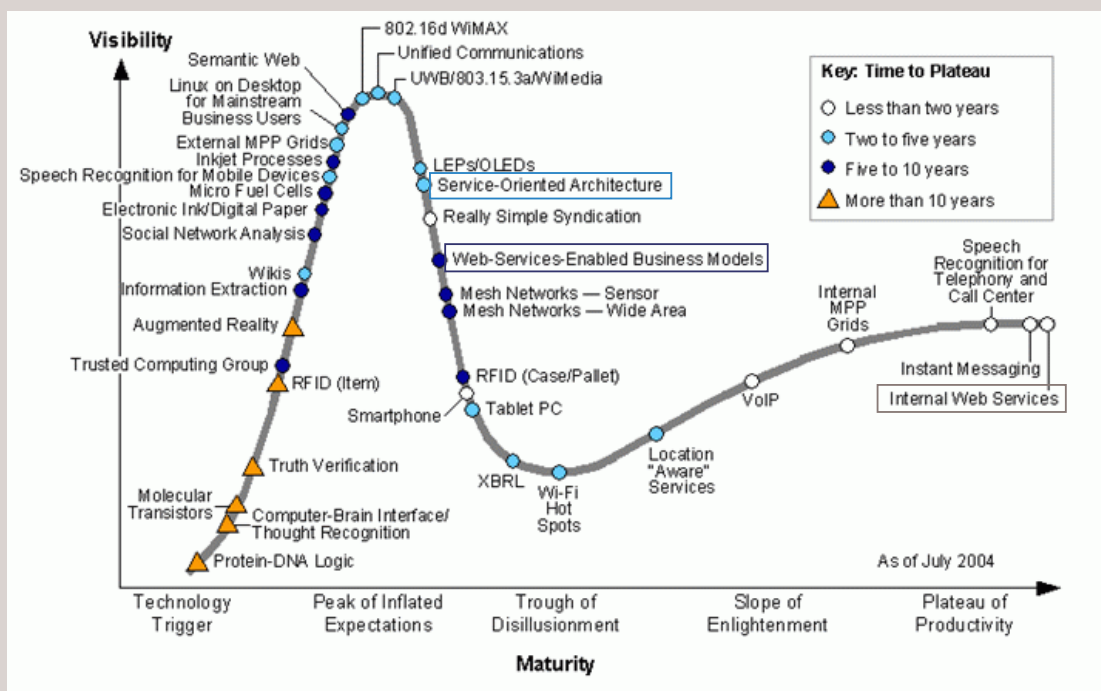
Schritt 1: Dienst finden

- gegeben: *standardisierter* Web Service
- gesucht: Anbieter des Web Services, einschl. WSDL-Beschreibung
- Um Anbieter *automatisch* zur Laufzeit zu finden, muss eine von den folgenden Bedingungen erfüllt sein:
 - a) Web Service ist **kostenlos** und **unkritisch**, daher kein Vertrag mit Anbieter nötig
Beispiel: kostenlose, verzögerte Aktienkurse als unverbindliche Information
 - b) es **besteht bereits ein Vertrag** mit dem Anbieter
Beispiel: SAP/R3-Web-Service auf unterschiedlichen unternehmenseigenen Servern

Web Services: Vision und Realität



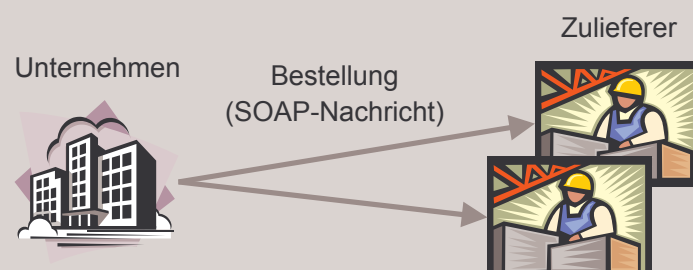
Web Services: Die Realität



Quelle: Gartner 2004

Anforderungen der Praxis

Beispiel



- Absender eindeutig identifizierbar?
- Nachricht unversehrt?
- Nachricht genau einmal übermittelt?
- Antwort nach bestimmter Zeit garantiert?
- Passt die Nachricht in den Geschäftsprozess?
z.B. keine Bestellung ohne vorherige Bestellanfrage

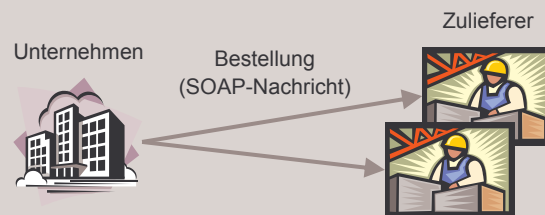
- Absender eindeutig identifizierbar?
- Nachricht unversehrt?
- Nachricht *genau einmal* übermittelt?
- Antwort nach bestimmter Zeit garantiert?
- Passt die Nachricht in den Geschäftsprozess?

- *keine* der Anforderungen von SOAP direkt unterstützt
- können aber als Zusatzinformationen im Briefkopf der SOAP-Nachricht realisiert werden

- digitale Signatur
 - Identifikation des Absenders
 - Unversehrtheit der Nachricht
- Anforderung einer Empfangsbestätigung
 - Nachricht mindestens einmal zugestellt
- eindeutige Nachrichtenreferenz
 - Erkennung von Duplikaten
- Verweis auf vorherige Nachrichten, z.B. Bestellanfrage
 - Berücksichtigung des Geschäftsprozesses (Workflows)

- Absender eindeutig identifizierbar?
- Nachricht unversehrt?
- Nachricht *genau einmal* übermittelt?
- Antwort nach bestimmter Zeit garantiert?
- Passt die Nachricht in den Geschäftsprozess?

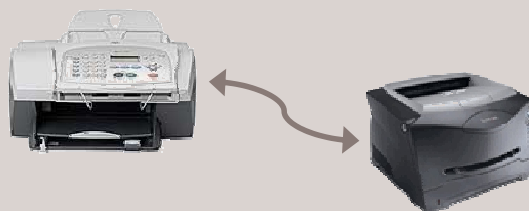
So einfach ist es dann doch nicht...



Problem der **Interoperabilität** (Zusammenarbeitfähigkeit):

- Zulieferer soll ersetzt werden
- Neuer Zulieferer bekommt WSDL-Beschreibung.
- WSDL beschreibt zwar *Syntax* der Schnittstelle, einschl. der Zusatzinformationen im Briefkopf.
- Was die Zusatzinformationen *bedeuten*, muss aber zwischen Unternehmen und Zulieferer geklärt werden.
- Grund: noch kein etablierter Standard hierfür

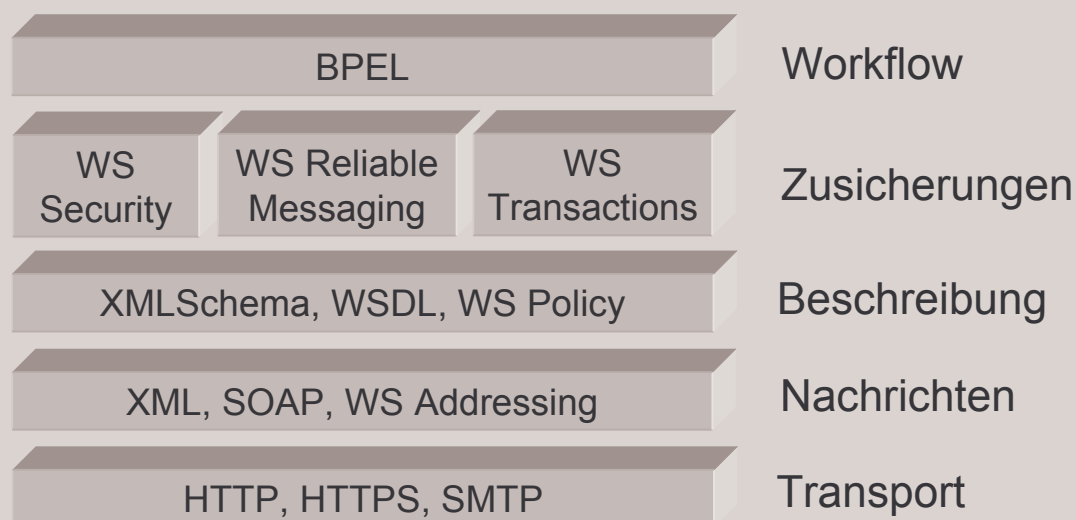
Interoperabilität



- Was interoperabel bedeutet, kann am Beispiel Fax verdeutlicht werden.
- Fax tatsächlich uneingeschränkt interoperabel:
- Faxgerät kann durch ein anderes ersetzt werden, *ohne* mit potentiellen Sendern Protokolle auszuhandeln.
- Z.B. muss *nicht* geklärt werden, wie Empfang eines Fax bestätigt wird.
- Grund: Es gibt einen etablierten internationalen Fax-Standard.

Erweiterungen von SOAP/WSDL

Standardisierung von Erweiterungen

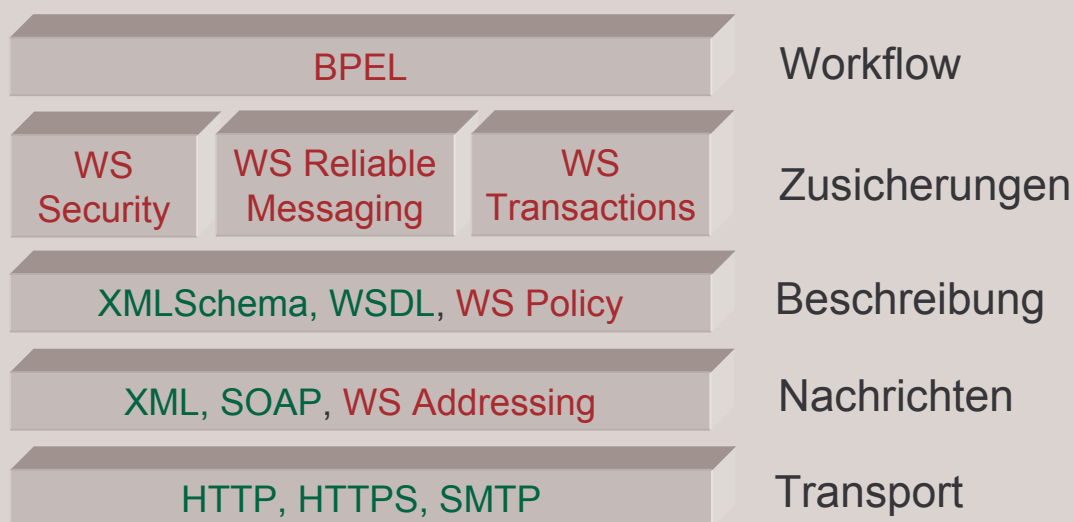


- sehr gute Übersicht:
<http://www-306.ibm.com/software/solutions/webservices/pdf/SecureReliableTransactedWSAction.pdf>

```

<S:Envelope ... >
  <S:Header>
    <wsa:ReplyTo>
      <w sa:Address>http://business456.com/User12</w sa:Address>
    </w sa:ReplyTo>
    <w sa:To>http://fabrikam123.com/Traffic</w sa:To>
    <w sa:Action>http://fabrikam123.com/Traffic/Status</w sa:Action>
    <w ssec:Security>
      <w ssec:BinarySecurityToken
        Value Type="w ssec:X509v3"
        Encoding Type="w ssec:Base64Binary">
        dWJzY3JpYmVyLVBic.....eFw0wMTEwMTAwMD
      </w ssec:BinarySecurityToken>
    </w ssec:Security>
    <w srm:Sequence>
      <w su:Identifier>http://fabrikam123.com/seq1234</w su:Identifier>
      <w srm:MessageNumber>10</w srm:MessageNumber>
    </w srm:Sequence>
  </S:Header>
  <S:Body>
    <app:TrafficStatus
      xmlns:app="http://highwayman.org/payloads">
      <road>520W</road> <speed>3MPH</speed>
    </app:TrafficStatus>
  </S:Body>
</S:Envelope>
  
```

Etablierte Standards



- etablierte Standards in grün
- (noch) nicht etablierte Standards in rot

Bewertung der Erweiterungen



- noch *keine* etablierten Standards
- + einzelne Erweiterungen *unabhängig* voneinander
- + meist gemeinsame Vorschläge von Microsoft und IBM

Keine *prinzipiellen* Hürden, jedoch noch ein *langer* Standardisierungsweg zu gehen!

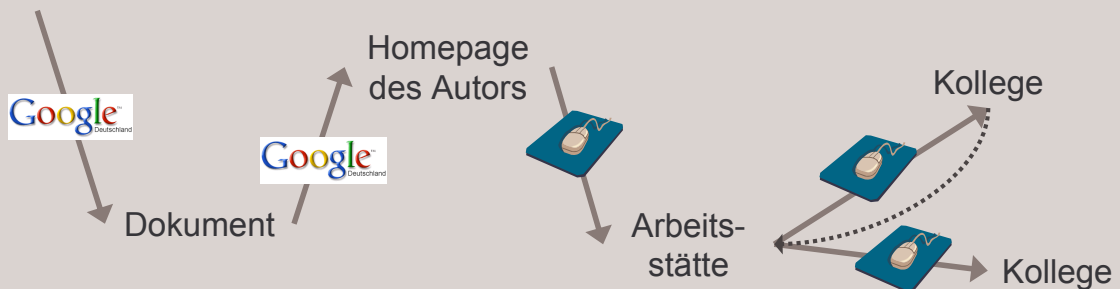
Semantic Web



Semantic Web: Berner-Lees` Vision



- Webinhalte und ihre Vernetzung werden für Maschinen verständlich.
- Komplexe Anfragen können ans Web gestellt werden, die heute nur durch Surfen zu beantworten sind.
- Beispiel: Mit welchen Kollegen arbeitet der Autor eines bestimmten Dokumentes zur Zeit zusammen?



Erste Schritte zum Semantic Web



1. XML + Namensräume + XML-Schema

- WWW als Menge von verlinkten, hierarchisch strukturierten Daten
- Vokabular zur Strukturierung definierbar
- Bedeutung dieses Vokabulars mit Namensräumen festgelegt

⇒ maschinenverarbeitbare verlinkte Daten, Links jedoch nicht maschinenverarbeitbar

2. RDF + Namensräume + RDF-Schema

- WWW als Menge vernetzter Ressourcen
- Vokabular für Beziehungen zwischen Ressourcen definierbar
- Bedeutung dieses Vokabulars mit Namensräumen festgelegt

⇒ maschinenverarbeitbares
Netzwerk von Beziehungen

Resource Description Framework

- W3C-Standard von 2004
- verschiedene Versionen:
 - Tripel: kompakt, lesbar
 - RDF/XML: für maschinelle Verarbeitung
- Tripel setzen bel. Webressourcen miteinander in Beziehung, z.B.:
 $\langle \text{URI-1, ist-Kollege-von, URI-2} \rangle$
- Triple können an bel. Stelle (URI) abgespeichert sein:
 $\text{URI: } \langle \text{URI-1, Relation, URI-2} \rangle$

Beispiel



"HTML-Dokument
index.html

wurde erstellt von

Person 85740"

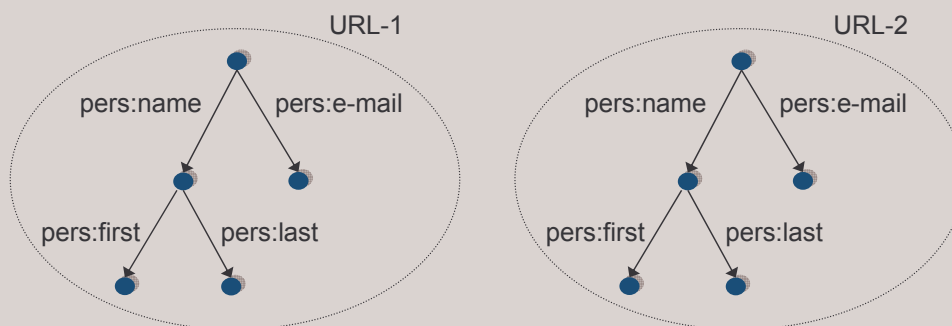


- Ressourcen und Beziehungen mit URIs identifiziert
- Falls ein Programm den Namensraum kennt, kennt es auch die Bedeutung der Ressource/Beziehung.
- weitergehende Infos: <http://www.w3.org/TR/rdf-primer/>

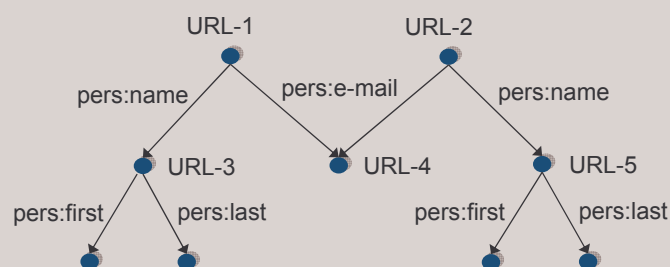
XML vs. RDF



XML



RDF



XML



- Bäume
- Kanten geordnet
- nicht verteilt

→ Struktur

XML = Spezialfall von RDF

RDF



- gerichtete Graphen
- Kanten ungeordnet
- verteilt

→ Inhalt

Semantic Web: Die Realität

XML

- XML heute omnipräsent, wenn auch nicht immer sichtbar. ✓

RDF

- HTML-Seiten und XML-Dokumente werden erstellt, aber kein RDF. Wozu auch?
- Hat Google Interesse an einem Semantic Web?
- neue XHTML-Version wird RDF integrieren:
Jedes XHTML-Element kann RDF-Meta-Informationen haben. Aber woher kommen diese Meta-Informationen?

⇒ für RDF bisher keine Killerapplikation

Rückblick

XML

Lernziele

- Was ist eine generische Auszeichnungssprache?
- Vor- und Nachteile einer generischen Auszeichnungssprache
- XML als Vereinfachung von SGML, die für das Web hinreichend allgemein ist.
- Grundbausteine und Syntaxregeln von XML
- SGML, HTML, XML und XHTML unterscheiden
- Welche Anwendungen von XML gibt es?

Lernziele

- Zwei Funktionen von Namensräumen:
 - Namenskonflikte auflösen
 - Bedeutung von Elementen und Attributen festlegen
- Namensräume werden in XML mit URIs bezeichnet
- Wie werden Elementen/Attributen in XML Namensräume zugeordnet?

Lernziele

- Wohlgeformtheit von Gültigkeit unterscheiden
- DTDs und XML-Schemata lesen können.
- Vorteile von XML-Schemata gegenüber DTDs
- Was ist ein Ziel-Namensraum?
- Wie wird aus einem XML-Dokument eine Instanz eines XML-Schemas?
- Was bedeutet Typsubstitution in Schema-Instanzen und welche Probleme ergeben sich hierdurch?
- Namensraum für Schema-Instanzen als Beispiel einer Erweiterung von XML

Lernziele

- Wie arbeitet ein SAX-Parser?
- Was ist ein DOM-Parser?
- Vor- und Nachteile von SAX- und DOM-Parser
- Welchen Parser für welche Anwendung?

Lernziele

- Wie arbeitet ein XSLT-Prozessor?
- Welche vordefinierten Templates gibt es?
- Zwei Arten, ein XSLT-Stylesheet zu verarbeiten: Vor- und Nachteile
- Vor- und Nachteile von XSLT

Lernziele

- drei Arten, XML persistent zu speichern
- Was muss beachtet werden, wenn XML als Speicherformat benutzt wird?
- Wie können Primär- und Fremdschlüssel in einem XML-Schema definiert werden?
- XML als Datenmodell oder relationales Datenmodell: Vor- und Nachteile

Lernziele

- Was sind Web Services?
- Was ist neu an Web Services und was nicht?
- Wichtigsten Anwendungen von Web Services
- RPCs vs. Messaging: Vor- und Nachteile

Lernziele

- Aufbau einer SOAP-Nachricht
- Wie werden RPCs in SOAP-Nachrichten kodiert?
- Wie wird sinnvollerweise ein SOAP-Nachrichtenformat erweitert?
- Was ist ein Kodierungsstil in SOAP?

Lernziele

- Grundidee und Struktur von WSDL
- Google-WSDL lesen und erweitern können (ohne Definition komplexer Datentypen)
- REST-kompatible SOAP-Bindungen

Lernziele

- Einbindung eines Web Services zur Laufzeit oder zur Entwicklungszeit:
Praxisrelevanz einschätzen
- Beschränkungen von SOAP/WSDL

Lernziele

- Unterschiede zwischen Datenmodell von XML und RDF

Wie geht es weiter?



heutige Übung

- Google-WSDL erweitern (keine Folien hierzu)
- Gelegenheit, Fragen zu stellen!

nächste Woche

- Klausur

Noch freie Plätze im WS 2005/6!



P Kundenprojekt Web-Technologien

- Projekt, 4 SWS, 8 Credits
- Plenum Mittwochs 12-16 Uhr
- Projektarbeit, wie in der Berufspraxis üblich
 - *keine* wohldefinierten Arbeitspakete
 - realer Kunde aus Industrie
 - Teilnehmer stellen Projektleitung

K Existenzgründungen in der IT-Industrie

- Kurs, 2 SWS, 4 Credits, ABV
- Mittwochs 10-12 Uhr
- mit Businessplan-Wettbewerb mit externen Gutachtern

Termin: Mittwoch den 13.7.2005

Zeit: 12:00-13:45 s.t.

Bitte unbedingt pünktlich um 12:00 s.t. eintreffen!

Raum:

- Nachnamen A-N: Takustr. 9, Vorlesungssaal 003
- Nachnamen O-Z: FB Physik, Arnimallee 14, Trakt 3, 1. Stock, Hörsaal A 1.3.14

Teilnahmevoraussetzungen:

- Anmeldung im Online-KVV
- zusätzlich für alle Master- und Bachelor-Studierende: Unterschrift muss auf der Anmeldeliste bereits vorliegen
- Identifizierung mit Lichtbildausweis

Hilfsmittel: keine, außer leeres Papier, Stift (kein Bleistift) und Fremdwörterlexikon