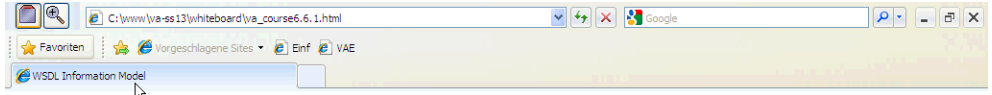
Script **generated by TTT**

Title: Distributed_Applications (25.06.2013)

Date: Tue Jun 25 14:30:37 CEST 2013

Duration: 88:04 min

Pages: 46



A WSDL document uses the following elements in the definition of network services:

Types: a container for non-built-in data type definitions using some type system, e.g. arrays and structures.

Message: an abstract, typed definition of the data being transferred between the requestor and service;

method call (request/response): modeled as 2 messages.

Port Type: an abstract set of operations supported by one or more endpoints; an operation specifies a specific input/output message sequence.

Operation: an abstract description of an action supported by the service.

Binding: specifies a concrete protocol and data format for the operations and messages defined by a particular PortType, such as SOAP or Corba.

Port: a single endpoint defined as a combination of a binding and a network address.

Service: a collection of related endpoints.

Parts of WSDL

Relationship of parts

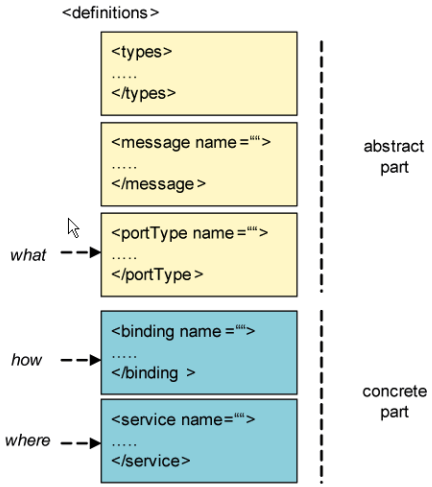
Generated by Targeteam

Parts of WSDL

WSDL is divided in 2 parts

an abstract part which describes what is offered; it consists of types, message, operations and port types.

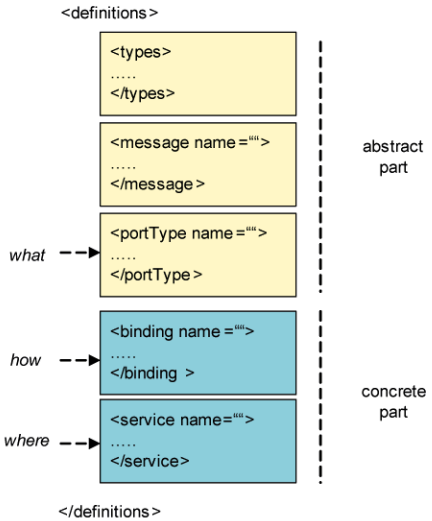
a concrete part which describes how and where it is offered; it consists of bindings, services and ports.



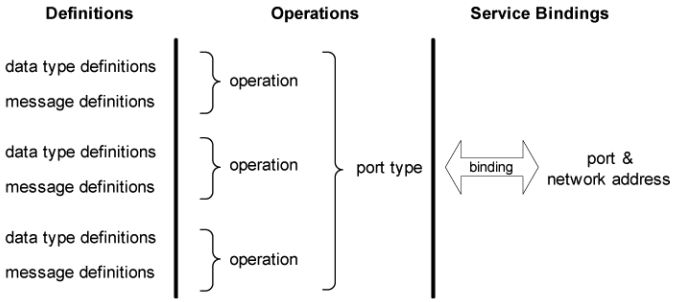
Parts of WSDL

an abstract part which describes what is offered; it consists of types, message, operations and port types.

a concrete part which describes how and where it is offered; it consists of bindings, services and ports.



Relationship of parts



definitions are generally expressed in XML.

operations describe actions for the messages supported by a Web Service; the equivalent of a method signature in Java.

service bindings connect port types to a port.

Generated by Targeteam

Example for SOAP Request/Response

WSDL definition of a simple service providing stock quotes; the service supports the single operation GetLastTradePrice(ticker symbol) and returns the price as a float.

```
<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"
    xmlns:xsd1="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/">

<types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
        <element name="TradePriceRequest">
            <complexType>
                <all><element name="tickerSymbol" type="string"/></all>
            </complexType>
        </element>
        <element name="TradePrice">
            <complexType>
                <all><element name="price" type="float"/></all>
            </complexType>
```

Example for SOAP Request/Response

```
            <complexType>
                <all><element name="tickerSymbol" type="string"/></all>
            </complexType>
        </element>
        <element name="TradePrice">
            <complexType>
                <all><element name="price" type="float"/></all>
            </complexType>
        </element>
    </schema>
</types>

<!-- Parameter der Nachricht -->
<message name="GetLastTradePriceInput">
    <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<!-- Parameter der Antwort -->
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
</message>

<portType name="StockQuotePortType">
```

Example for SOAP Request/Response

```
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>

<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document"
```

Example for SOAP Request/Response

```
</message>

<!-- Parameter der Antwort -->
<message name="GetLastTradePriceOutput">
    <part name="body" element="xsd1:TradePrice"/>
</message>

<portType name="StockQuotePortType">
    <operation name="GetLastTradePrice">
        <input message="tns:GetLastTradePriceInput"/>
        <output message="tns:GetLastTradePriceOutput"/>
    </operation>
</portType>

<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
        <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
        <input><soap:body use="literal"/></input>
        <output><soap:body use="literal"/></output>
    </operation>
</binding>
```

operation is specified as a sequence of messages

Web Services Description Language (WSDL)

Ian Forster states: "Web service have little value if others cannot discover, access, and make sense of them."

Definition: A WSDL document defines **services** as collections of network endpoints, or ports.

WSDL has a purpose similar to that of IDLs in conventional middleware platforms. A WSDL description describes 3 fundamental properties of a Web Service

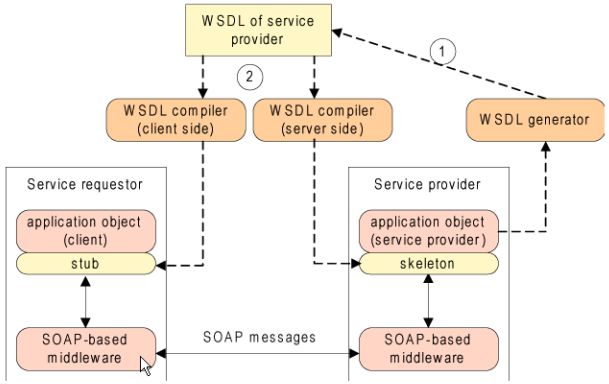
What a service does: operations and the arguments needed to invoke them.

How a service is accessed: details of data formats and protocols.

Where a service is located: details of the protocol-specific network address, such as a URI.

Generating code from WSDL

Use of a WSDL compiler to automatically generate code (e.g. a Java interface) from a WSDL file.



WSDL documents can be generated from APIs (1).

Stubs and skeletons can be generated from WSDL document (2).

Common bad Practices

Analysis of existing WSDL documents shows that functionality of many Web services are hard to understand due bad practices.

developers take not sufficient care of names and comments.

port types are tied to concrete protocols.

semantically unrelated operations are placed in a single port type.

overload output messages to transport results and error information.

provides the definition of a set of services supporting the description and discovery of

businesses, organizations, and Web Service providers,

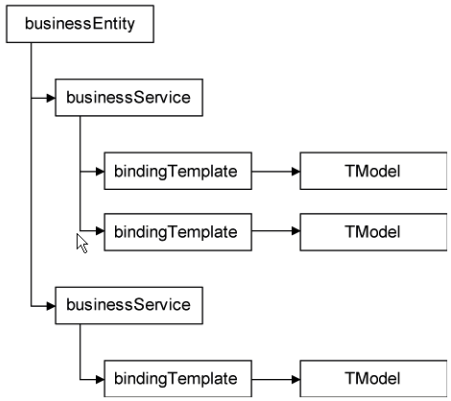
the Web services they make available,

the technical interface to access those services.

UDDI itself is a Web Service; has a WSDL interface and can be described by a UDDI registry.

UDDI registry xmethods for publicly available Web Services.

UDDI allows to store and manipulate four main types of entities



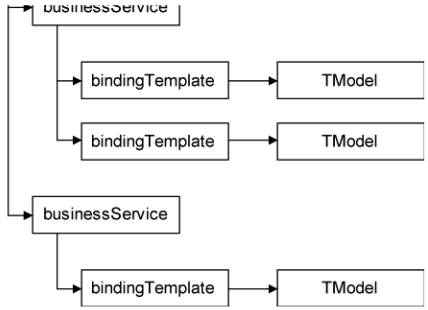
businessEntity : represents the owner of a Web Service.

Attributes: name, unique key, zero or more services, descriptions, ...

businessService : represents a group of one or more Web Services.

Attributes: name, unique key, one binding template per Web Service, descriptions, ...

bindingTemplate : represents a single Web Service; contains all the information to locate and invoke the service



businessEntity : represents the owner of a Web Service.

Attributes: name, unique key, zero or more services, descriptions, ...

businessService : represents a group of one or more Web Services.

Attributes: name, unique key, one binding template per Web Service, descriptions, ...

bindingTemplate : represents a single Web Service; contains all the information to locate and invoke the service

Attributes: unique key, an access point that indicates the URL of the Web Service

TModel : represents WSDL interface types.

Attributes: name, unique key, an URL that points to the data associated with the TModel, description, ..

UDDI registries have 3 main types of users

service providers that publish services

requesters that look for services

other registries that need to exchange information.

UDDI supports the following sets of APIs

UDDI Inquiry API : operations to find registry entries such as `find_service` , or get details on specific entity, e.g. `get_serviceDetail` .

UDDI Publishers API : add, modify, and delete entries, e.g. `save_service` or `delete_service` .

UDDI Security API : get and discard authentication tokens to be used in communication with registry.

UDDI Ownership Transfer API : transfer ownership of structures between registries.

UDDI Subscription API : enables monitoring of changes in a registry by subscribing to track new, modified, and deleted entries.

UDDI Replication API : supports replication of information between registries.

provides the definition of a set of services supporting the description and discovery of

businesses, organizations, and Web Service providers,

the Web services they make available,

the technical interface to access those services.

UDDI itself is a Web Service; has a WSDL interface and can be described by a UDDI registry.

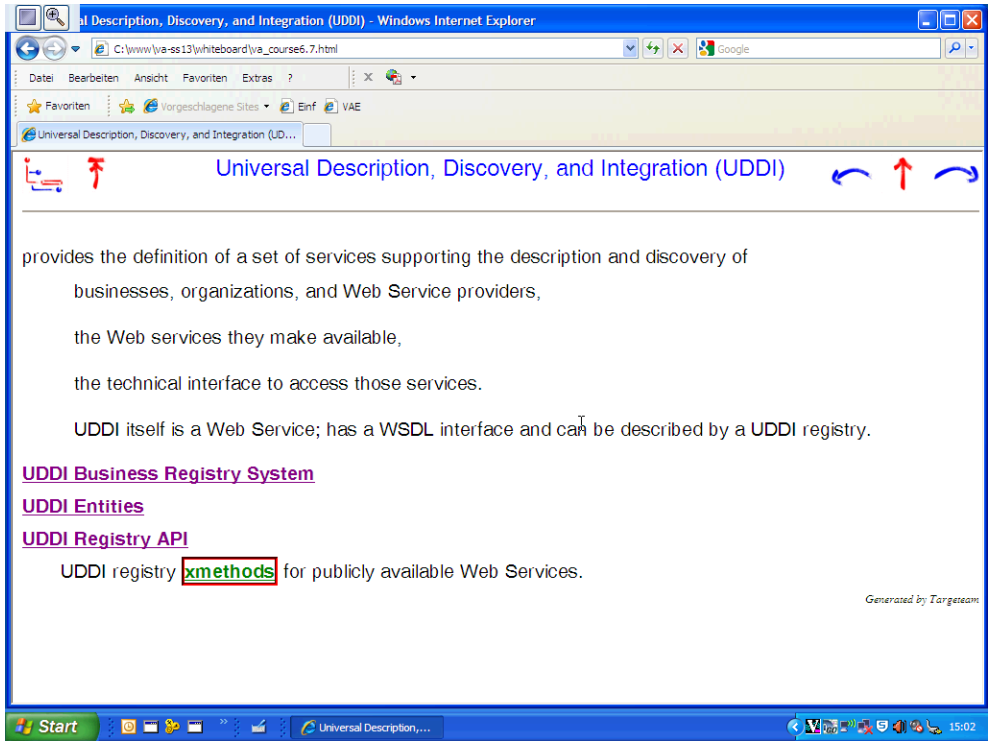
UDDI Business Registry System

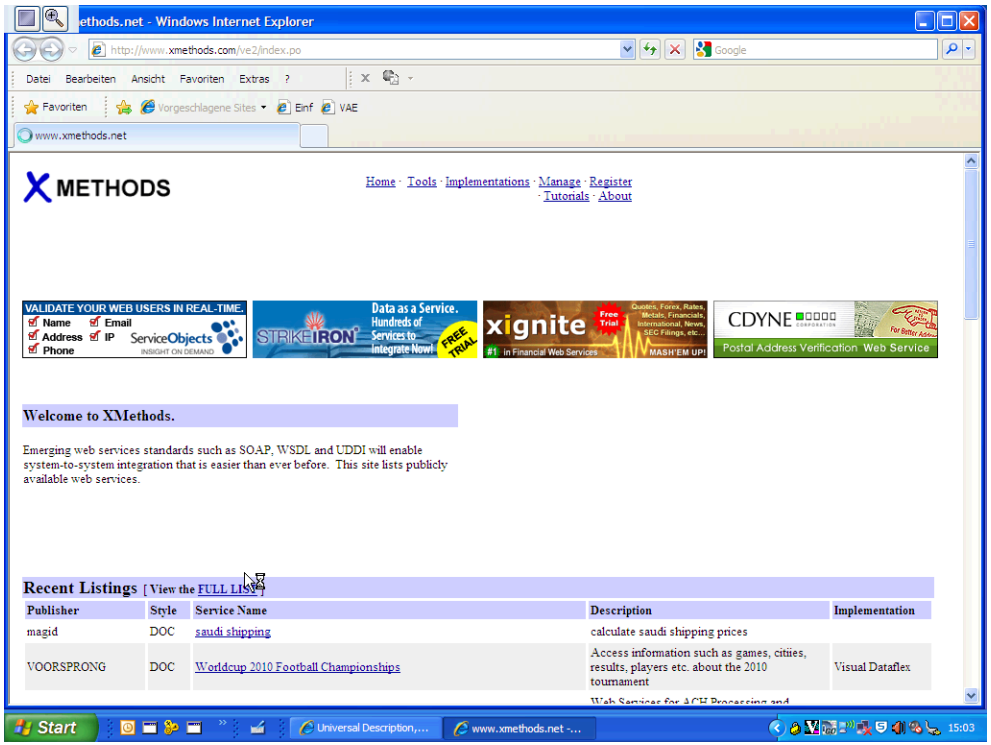
UDDI Entities

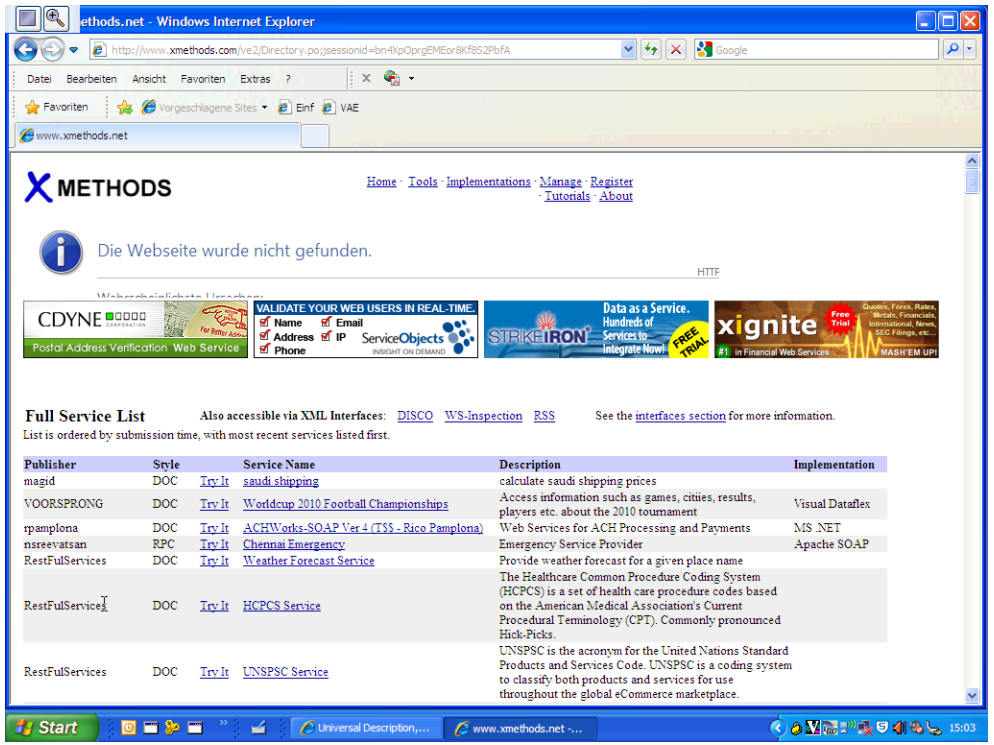
UDDI Registry API

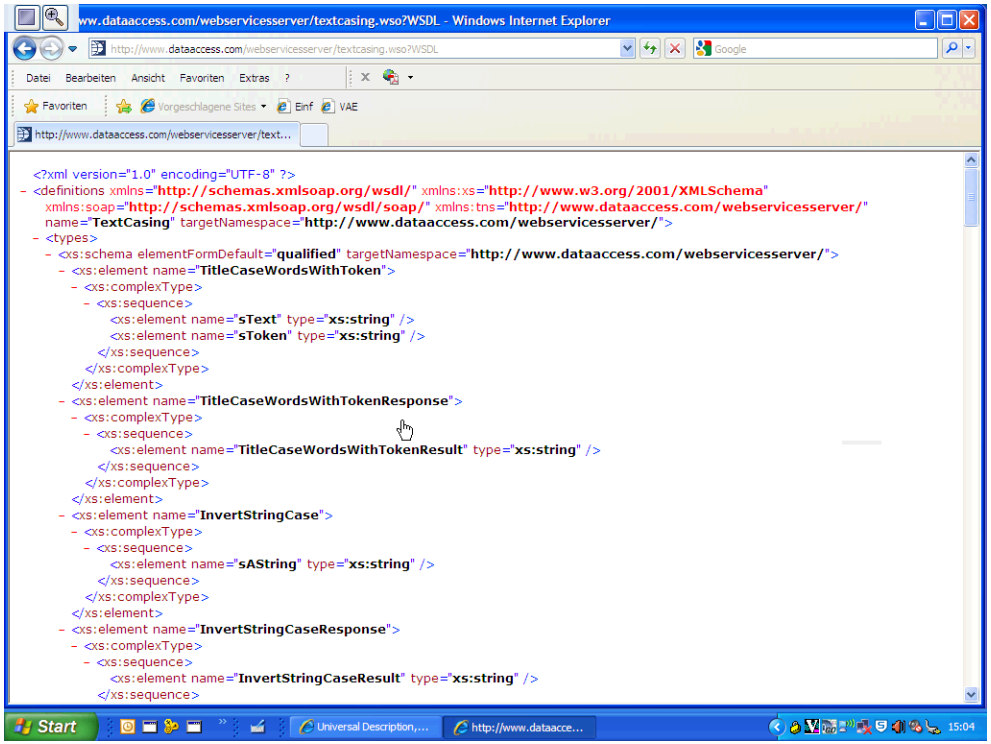
UDDI registry xmethods for publicly available Web Services.

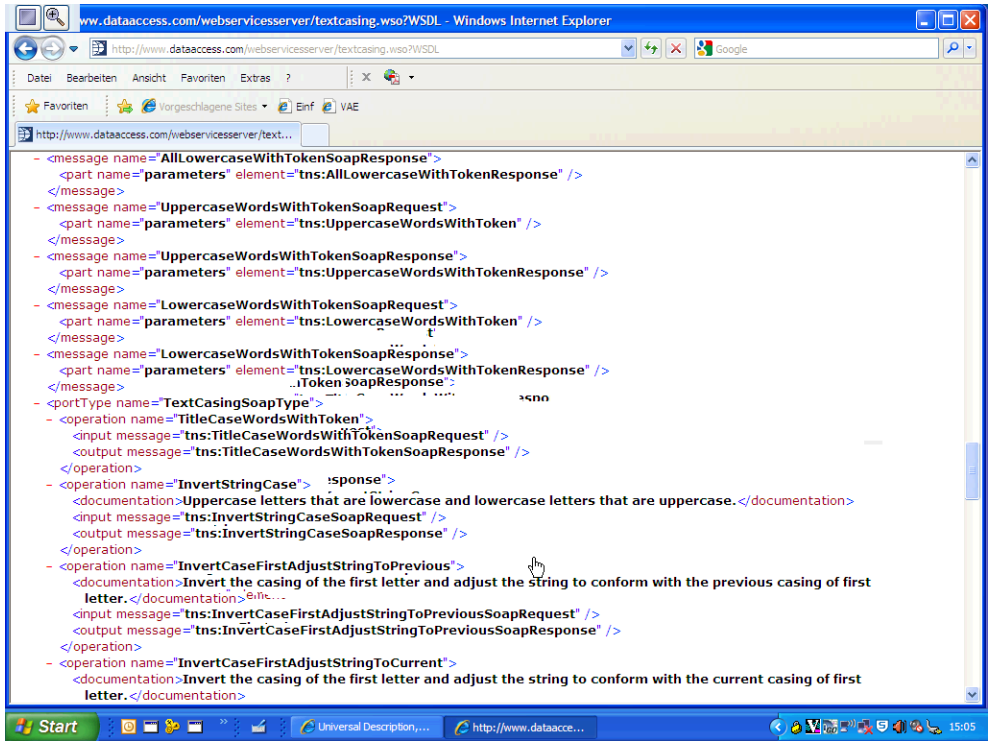
Generated by Targeteam

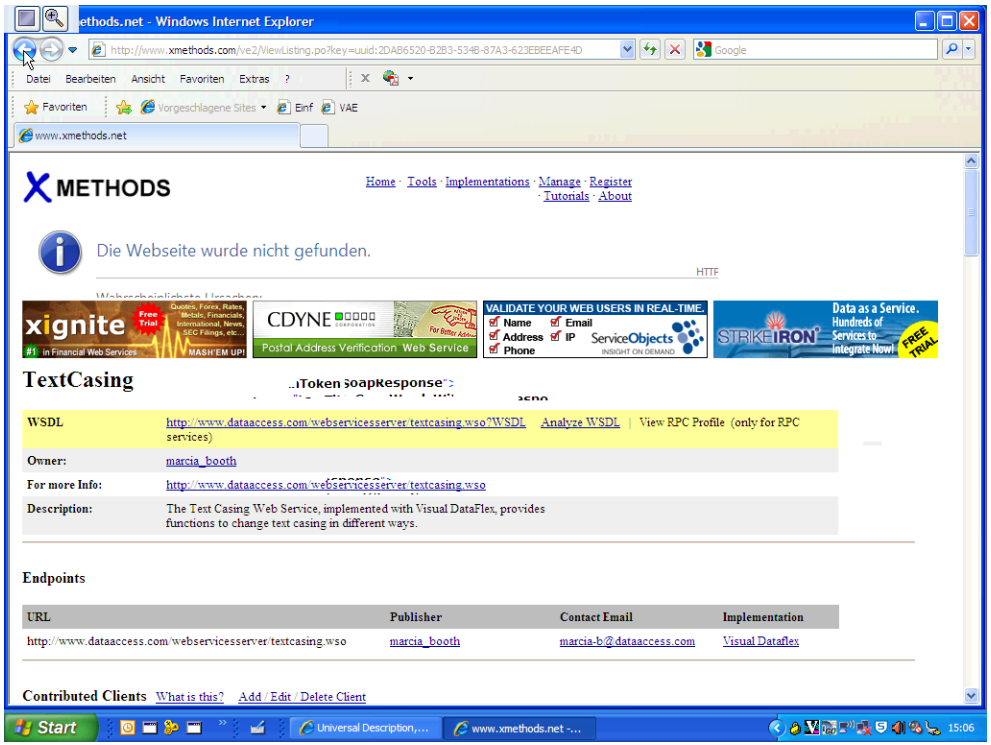


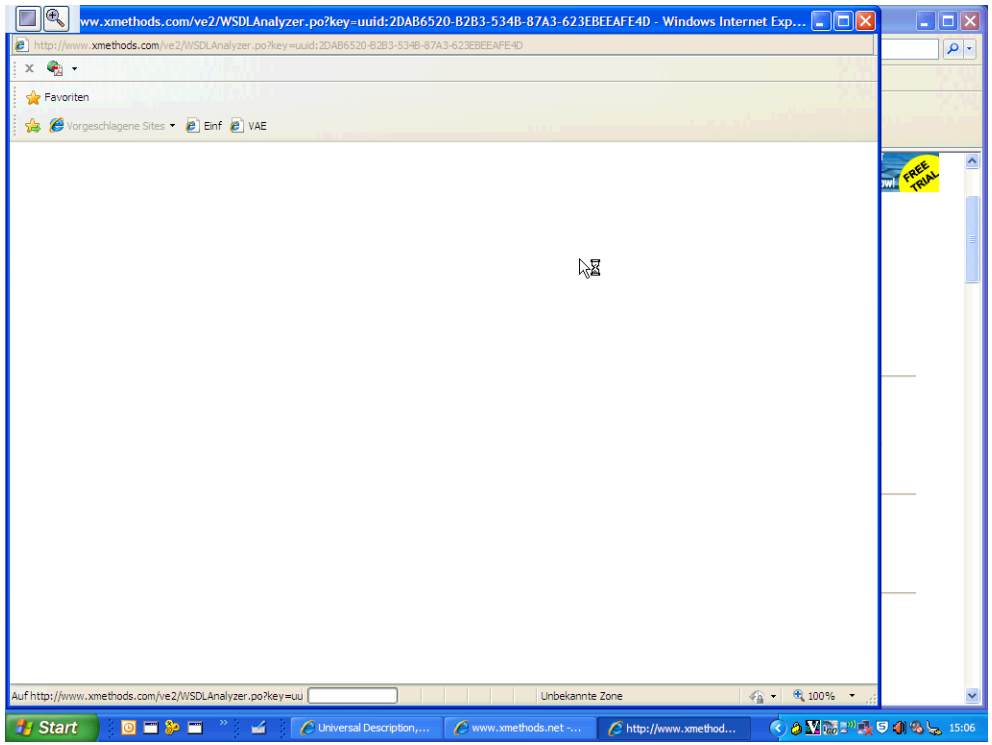


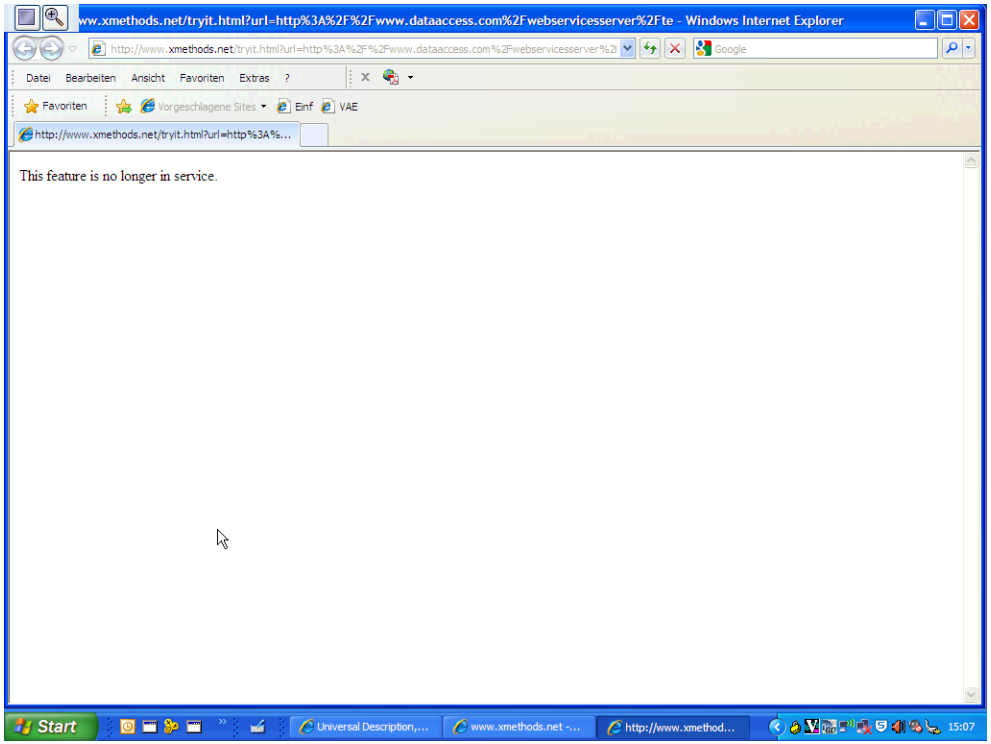



www.dataaccess.com/webservicesserver/textcasing.wso?WSDL - Windows Internet Explorer
<?xml version="1.0" encoding="UTF-8" ?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.dataaccess.com/webservicesserver/"
name="TextCasing" targetNamespace="http://www.dataaccess.com/webservicesserver/">
<types>
<xs:schema elementFormDefault="qualified" targetNamespace="http://www.dataaccess.com/webservicesserver/">
<xs:element name="TitleCaseWordsWithToken">
<xs:complexType>
<xs:sequence>
<xs:element name="sText" type="xs:string" />
<xs:element name="sToken" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="TitleCaseWordsWithTokenResponse">
<xs:complexType>
<xs:sequence>
<xs:element name="TitleCaseWordsWithTokenResult" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="InvertStringCase">
<xs:complexType>
<xs:sequence>
<xs:element name="sAString" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="InvertStringCaseResponse">
<xs:complexType>
<xs:sequence>
<xs:element name="InvertStringCaseResult" type="xs:string" />
</xs:sequence>


www.dataaccess.com/webservicesserver/textcasing.wso?WSDL - Windows Internet Explorer
<message name="AllLowercaseWithTokenSoapResponse">
<part name="parameters" element="tns:AllLowercaseWithTokenResponse" />
</message>
<message name="UppercaseWordsWithTokenSoapRequest">
<part name="parameters" element="tns:UppercaseWordsWithToken" />
</message>
<message name="UppercaseWordsWithTokenSoapResponse">
<part name="parameters" element="tns:UppercaseWordsWithTokenResponse" />
</message>
<message name="LowercaseWordsWithTokenSoapRequest">
<part name="parameters" element="tns:LowercaseWordsWithToken" />
</message>
<message name="LowercaseWordsWithTokenSoapResponse">
<part name="parameters" element="tns:LowercaseWordsWithTokenResponse" />
</message>
<portType name="TextCasingSoapType">
<operation name="TitleCaseWordsWithToken">
<input message="tns:TitleCaseWordsWithTokenSoapRequest" />
<output message="tns:TitleCaseWordsWithTokenSoapResponse" />
</operation>
<operation name="InvertStringCase">
<documentation>Uppercase letters that are lowercase and lowercase letters that are uppercase.</documentation>
<input message="tns:InvertStringCaseSoapRequest" />
<output message="tns:InvertStringCaseSoapResponse" />
</operation>
<operation name="InvertCaseFirstAdjustStringToPrevious">
<documentation>Invert the casing of the first letter and adjust the string to conform with the previous casing of first letter.</documentation>
<input message="tns:InvertCaseFirstAdjustStringToPreviousSoapRequest" />
<output message="tns:InvertCaseFirstAdjustStringToPreviousSoapResponse" />
</operation>
<operation name="InvertCaseFirstAdjustStringToCurrent">
<documentation>Invert the casing of the first letter and adjust the string to conform with the current casing of first letter.</documentation>


ethods.net - Windows Internet Explorer
METHODS
Home · Tools · Implementations · Manage · Register
· Tutorials · About
Die Webseite wurde nicht gefunden.
TextCasing
WSDL http://www.dataaccess.com/webservicesserver/textcasing.wso?WSDL Analyze WSDL | View RPC Profile (only for RPC services)
Owner: marcia_booth
For more Info: http://www.dataaccess.com/webservicesserver/textcasing.wso
Description: The Text Casing Web Service, implemented with Visual DataFlex, provides functions to change text casing in different ways.
Endpoints
URL Publisher Contact Email Implementation
http://www.dataaccess.com/webservicesserver/textcasing.wso marcia_booth marcia-b@dataaccess.com Visual Dataflex
Contributed Clients What is this? Add / Edit / Delete Client


ww.xmethods.com/ve2/WSDLAnalyzer.po?key=uuid:2DAB6520-B2B3-534B-87A3-623EBEEAFE4D - Windows Internet Exp...
Favoriten
Vorgeschlagene Sites
Einf
VAE



Web Services

Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardization by the W3C community.

Motivation - Example

Service Oriented Architecture - SOA

Web Services - Characteristics

Web Services Architecture

Simple Object Access Protocol (SOAP)

Web Services Description Language (WSDL)

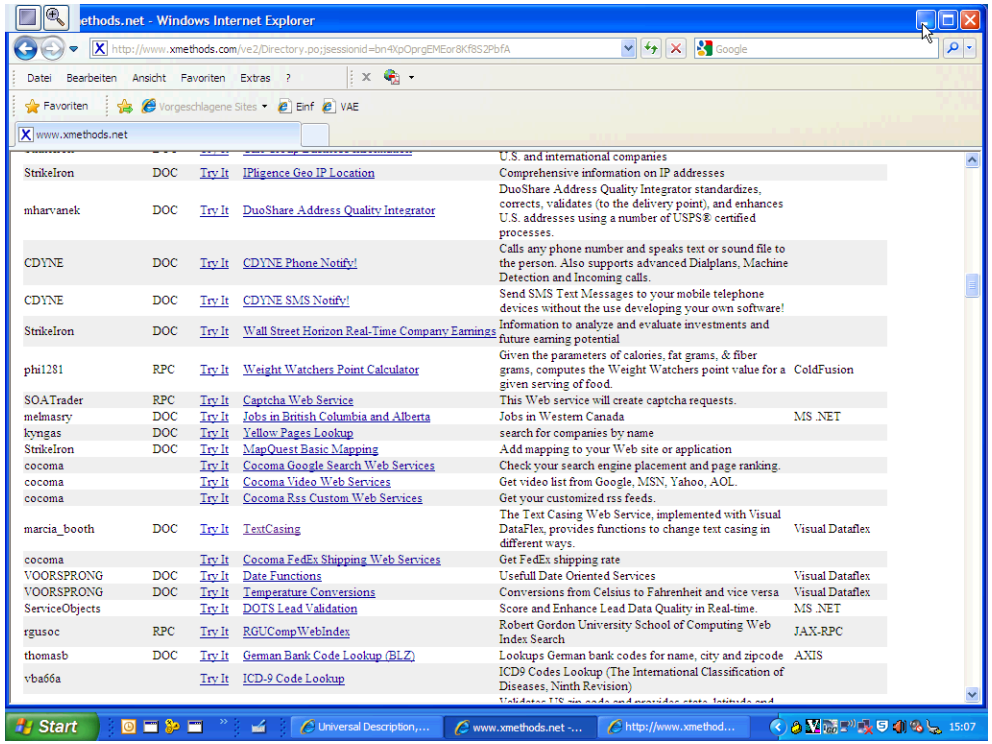
Universal Description, Discovery, and Integration (UDDI)

REST

Web Service Composition

Adopting Web Services

Mashups



REST

REST (Representational State Transfer) is an architectural style of distributed applications.

REST is not a standard; it is a set of principles how to use Web standards, such as HTTP, URIs and Mime Types.

The Web is a REST system.

REST is based on the following key principles

give every relevant resource an ID: use URIs to identify everything that is any item of interest.

URL: http://www.boeing.com/aircraft/747

A representation of the resource is returned (e.g., Boeing747.html). The representation places the client application in a state.

nk resources together: navigating links results in state transfers of the client application.

use standard methods: such as get, post, put, delete.

communication is stateless.

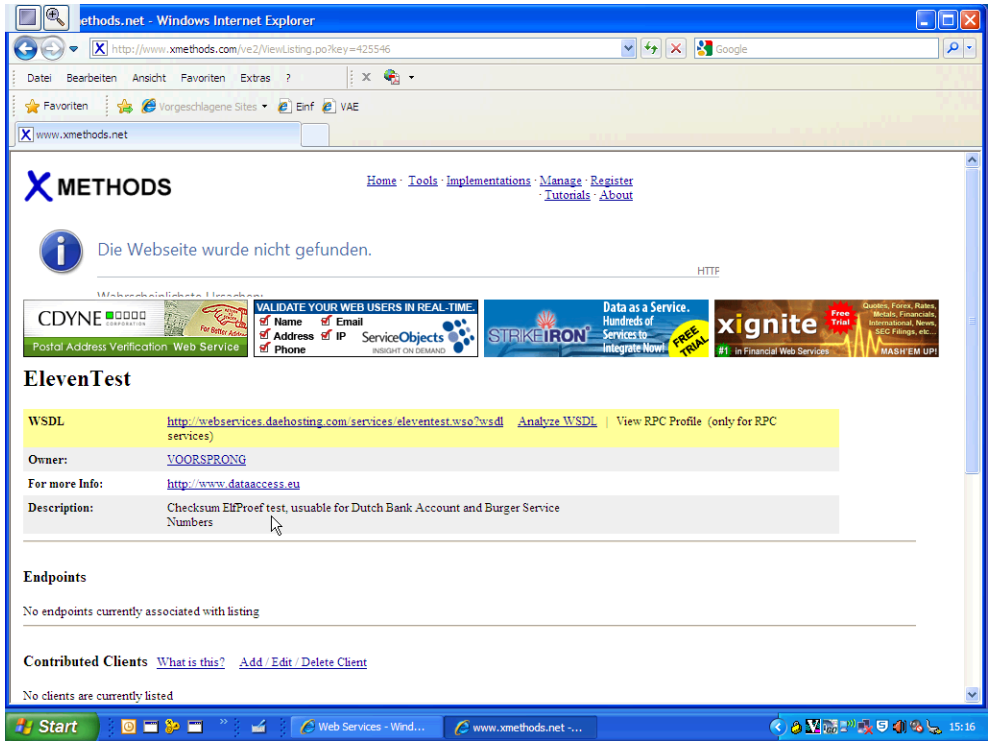
resources with multiple representations: client may specify the formats which it accepts

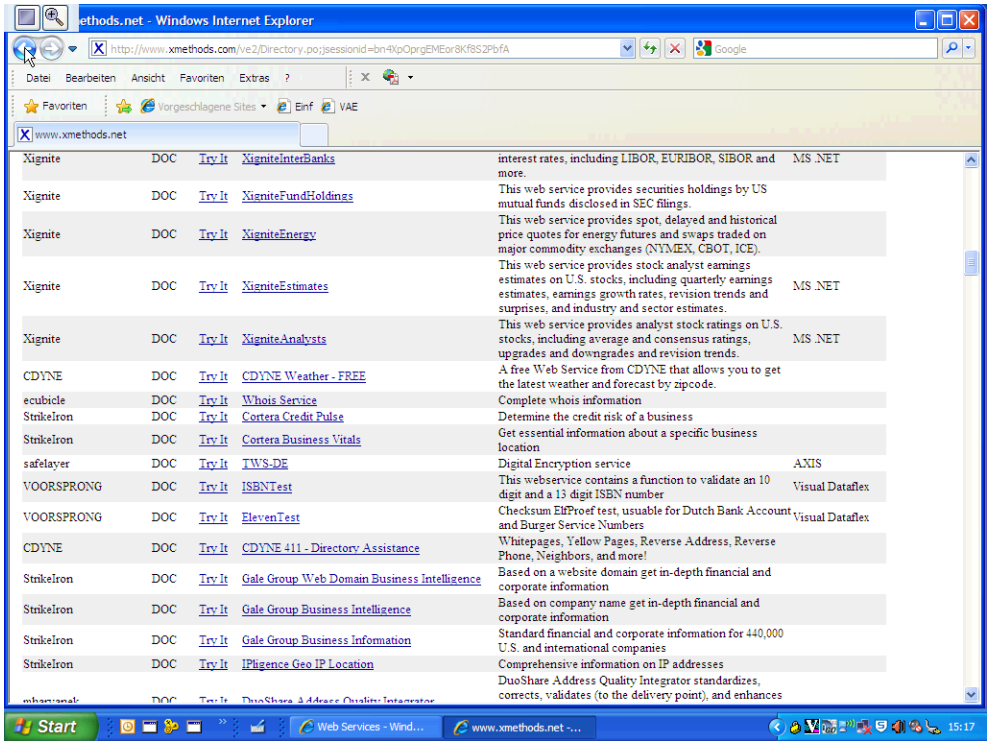
GET /customers/1234 HTTP/1.1

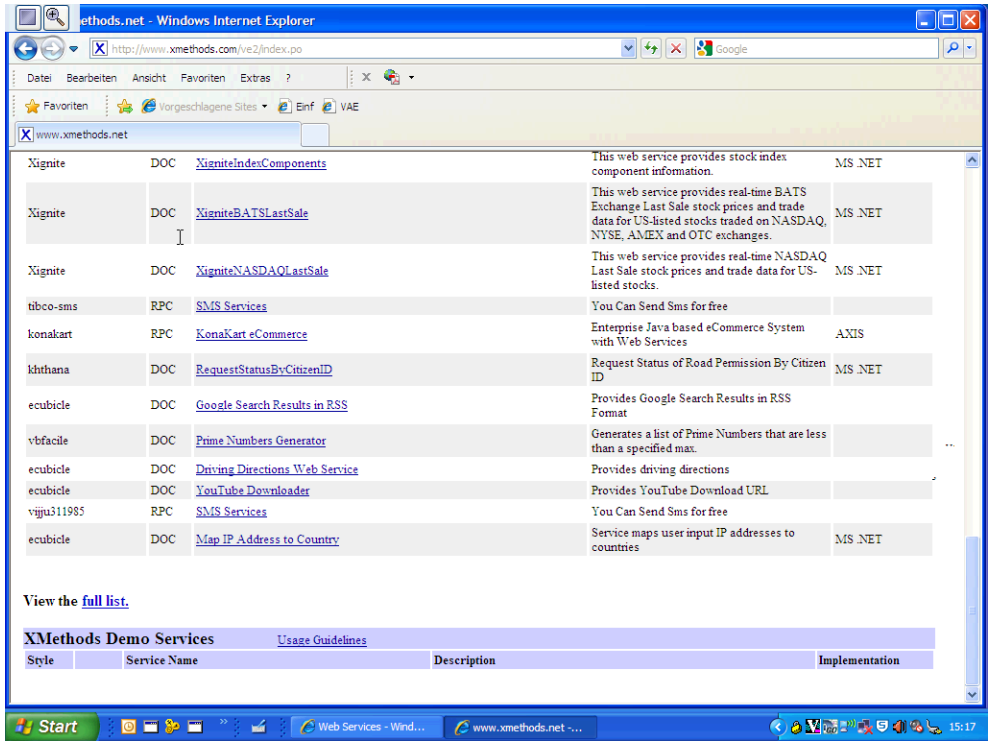
Accept: text/x-vcard

Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardization by the W3C community.

Generated by Targeteam







Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardization by the W3C community.

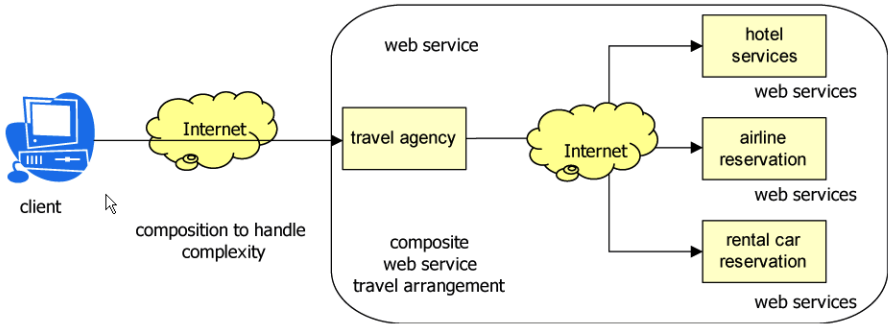
Web Service Composition

an important issue is the choice of the appropriate granularity

small vs. large Web Services - thousands vs. a handful of Web Services

what are the appropriate reusable, shared business components

Composition of complex Web Services from smaller reusable Web Services



Dimensions to handle complexity

Web Service Orchestration

Dimensions to handle complexity

component model: defines the sub-services.

orchestration model: defines the order in which the sub-services are invoked.

WS-Coordination is an extensible framework that describes how different Web Services work together reliably. Coordination framework contains

Activation, Registration and Coordination services

data access model: specifies the data exchange between the sub-services.

transactional model: transactional semantics of the composed service.

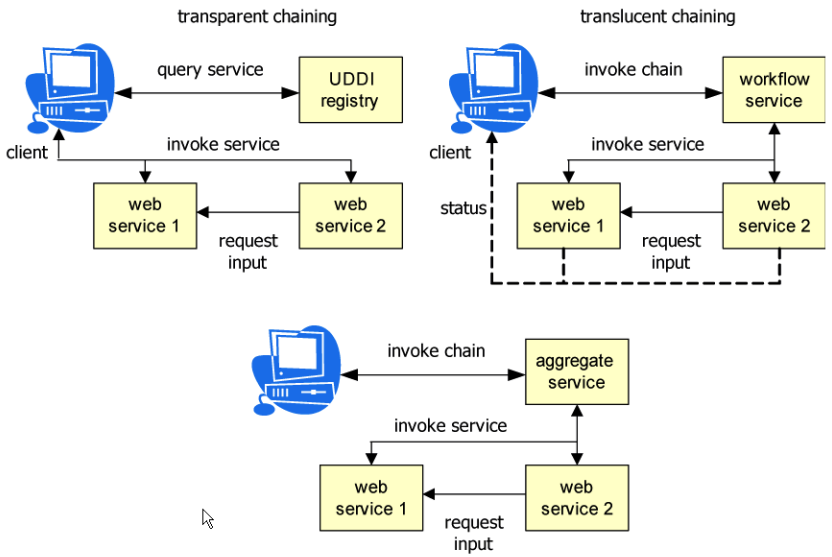
WS-Transaction specifies the protocols for each coordination type (used by WS-Coordination)

AtomicTransactions: all-or-nothing property, 2-phase-commit.

Business Activity: handle long-lived activities and to apply business logic to handle business exceptions; BusinessAgreement Protocol.

exception handling: handling of errors in the sub-services.

Web Service Orchestration



transparent chaining

Web Services

Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardization by the W3C community.

Motivation - Example

Service Oriented Architecture - SOA

Web Services - Characteristics

Web Services Architecture

Simple Object Access Protocol (SOAP)

Web Services Description Language (WSDL)

Universal Description, Discovery, and Integration (UDDI)

REST

Web Service Composition

Adopting Web Services

Mashups

Generated by Targeteam

Example Web Services

The following lists some available Web services; often registration necessary in order to use them.

Amazon E-Commerce Service (ECS)

FedEx Office and Priniting Service

printing of online documents and distribution of paper documents as commercial Web Service

free print plug-in for standard office application; plug-in added to list of printers

Pick up your document at any U.S. location or ship via FedEx for added convenience.

location independent use of printing service

Via Michelin

Reverse Geocoding Web Service allows users to obtain the closest road segment (named or not) for each supplied geographic coordinates (WGS84).

XMethods : clearinghouse for Web Services

Generated by Targeteam

Amazon E-Commerce Service (ECS)

ECS provides access to Amazon's product database with the following types of data

detailed product information.

customer-contributed content, e.g. wish list, product reviews.

seller information.

3rd party product information.

shopping cart contents.

ECS supports both SOAP and REST style interactions.

product operations

ItemSearch: performs a search for a specific item, typically using a set of keywords

SimilarityLookup: returns a list of similar products to a given product-ID (based product specifications and features).

ItemLookup: access to the data related to a specific product.

remote shopping cart operations

CartCreate: creates a remote shopping cart.

CartAdd: adds an item to the shopping cart.

further operations are CartGet (obtains content of the cart), CartModify (remove an item from cart)

Example Web Services

The following lists some available Web services; often registration necessary in order to use them.

Amazon E-Commerce Service (ECS)

FedEx Office and Priniting Service

printing of online documents and distribution of paper documents as commercial Web Service

free print plug-in for standard office application; plug-in added to list of printers

Pick up your document at any U.S. location or ship via FedEx for added convenience.

location independent use of printing service

Via Michelin

Reverse Geocoding Web Service allows users to obtain the closest road segment (named or not) for each supplied geographic coordinates (WGS84).

XMethods : clearinghouse for Web Services

Generated by Targeteam

Apache Axis

Apache Axis supports an environment to implement and provide Web services.

- set of client-side APIs for dynamically invoking SOAP Web services (with or without WSDL descriptions).
- tools to translate WSDL documents into Java frameworks.
- mechanisms for hosting Web services either within a servlet container (e.g. Tomcat) or via standalone server.
- a set of APIs for manipulating SOAP envelopes, bodies, and headers, and using them inside Message objects.
- data binding which enables mapping of Java classes into XML schemas and vice versa.
- a transport framework that allows usage of a variety of underlying transport mechanisms (e.g. JMS, email, etc).

Axis2

In the meantime there exists already Apache Axis2

- Java-based implementation of both the client and server sides of the Web services
- Axis2 is more flexible, efficient and configurable in comparison to Axis1.x
- Axis2 not only supports SOAP messages, but it also supports RESTful Web services.

Web Services and Java

Java provides a number of APIs implementing the Web Services standards

SAAJ (SOAP with Attachments API for Java)

- SOAP messages as Java objects

JAX-WS (Java API for XML based Web-Services)

- programming model for Web Services; replaces JAX-RPC

JJWSDL: Accessing WSDL descriptions

JAXR (Java API for XML Registries)

- Accessing Web Services Registries, e.g. UDDI

JAXP (Java API for XML Processing)

- Abstract XML-API-Specification implemented by e.g. Apache Xalan(XSLT), Apache Xerces2 (XML Parsing (DOM, SAX..)).

XWSS (Java Web-Services Security)

- Signatures, Encryption (roughly for SOAP what SSL is for HTTP)

Integration and WS Standards

<table>
<tr><td colspan="2">process management

process modeling, execution: BPEL4WS
process control : -
user interface integration : WSUI/WSXL</td><td rowspan="4">meta data & additional services

meta database : UDDI

system management : -

security services: WS-Security, SAML, XML-Encryption, XML-Signature

development support : -</td></tr>
<tr><td colspan="2">message management

transformation services : XSLT
synchronization : -
transaction services: WS-Coordination , WS Transactions</td></tr>
<tr><td>adapter

interface description: WSDL
messaging: SOAP</td><td>middleware

interface description : WSDL
reliability: WS Reliability
Messaging: SOAP , XML
Transport : HTTP, SMTP, ..</td></tr>
<tr><td colspan="2">physical network

transport layer: TCP, UDP
network layer: IP</td></tr>
</table>

Supporting - Restraining Forces

The adoption of Web Services in organization depends on

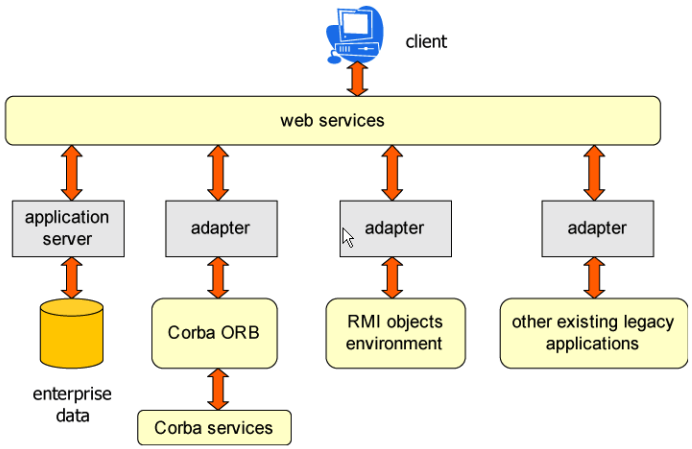
Supporting Forces

- interoperable networked applications, i.e. independence of hardware, operating system, application server, …
- easier exchange of distributed data
- easier access of enterprise wide data
- availability of external services, encapsulation of legacy applications
- cross-organizational computing
- reduced maintenance cost, easier reuse of components
- emerging industry-wide standard

Restraining Forces

- different formats and semantics of data sources
- security issues due to network access
- standards are evolving and not fixed yet
- lack of understanding of effects on operational systems

Distributed Process Architecture



Semantic Web Services

In order to allow for ***automatic discovery of appropriate*** web services and of ***automatic interaction / chaining / incorporation*** with web services

we need semantic meta-data for web services: Web-Service Ontologies, DataTypes with rich semantics....

Example: Map-Service

Input: (int, int)

Output: APPLICATION/GIF

Input: (int, int):

(x,y) of center of map ?

of corner of map ? which corner ?

what coordinate system? Wgs84? Gauss-Krueger? ...

Output: APPLICATION/GIF:

What kind of map? Topological? Political? POI? Traffic?

Units of measure?

candidate technology: OWL-S (Ontology Web Language for Web Services)

OWL-based Web service ontology, which supplies Web service providers with a core set of constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form.

Adopting Web Services

There exist already a variety of free of commercial Web services; provided especially by Internet companies, such as Google, Amazon or Yahoo.