Introduction to WSDL

WSDL

- **W**eb **S**ervice **D**efinition **L**anguage
- Predecessors include
  - COM, CORBA IDLs
  - Network Accessible Service Specification Language (IBM)
  - SOAP Contract Language (Microsoft)
  - First submitted to W3C in Sep 2000
  - Current version is 1.1

WSDL

- Define a web service in WSDL by
  - Writing an XML document conforming to the WSDL specs
- Describes three fundamental properties
  - What a service does
    - Operations (methods) provided by the service
  - How a service is accessed
    - Data format and protocol details
  - Where a service is located
    - Address (URL) details

Working of WSDL

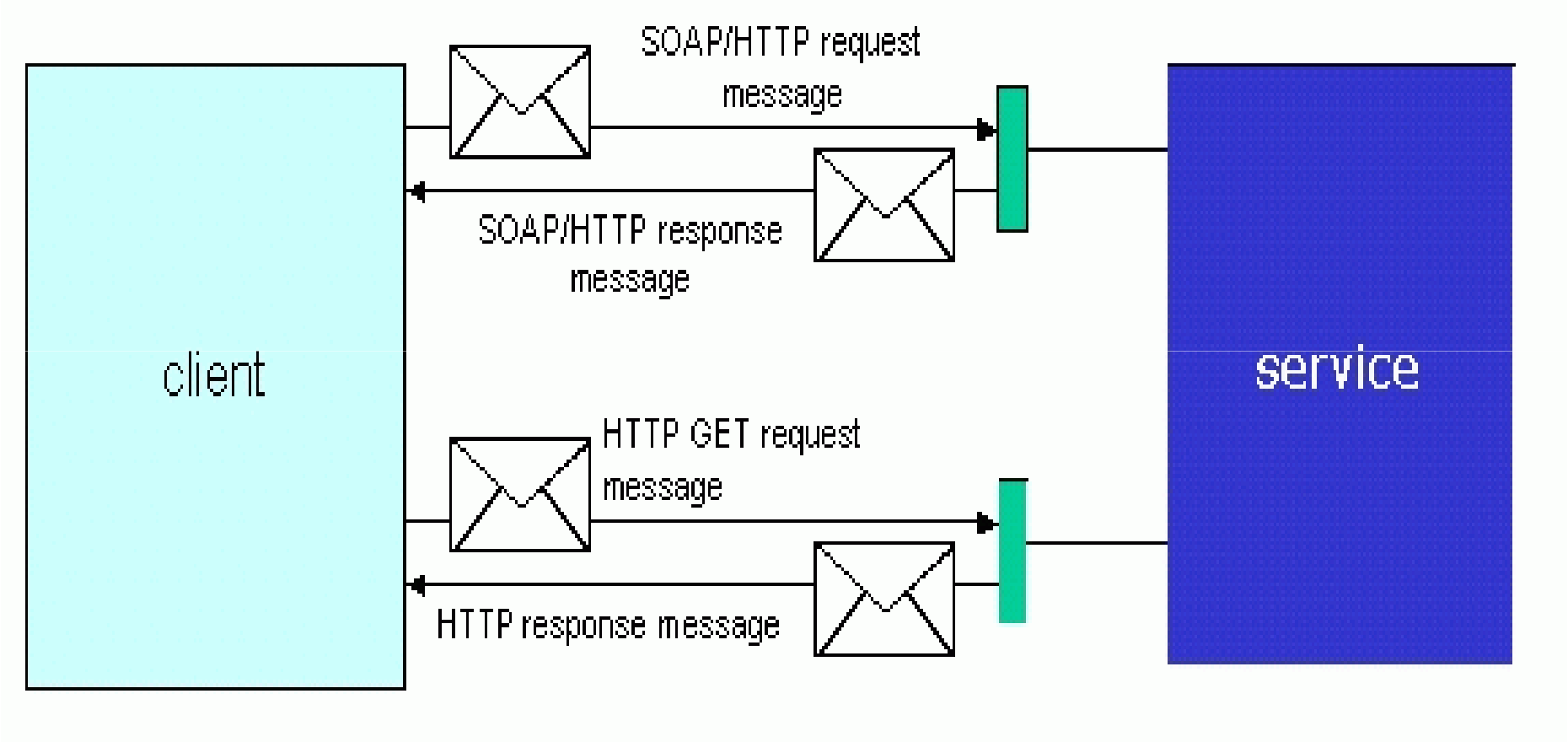


Figure 1. A client invoking a Web service.

Working of WSDL contd.

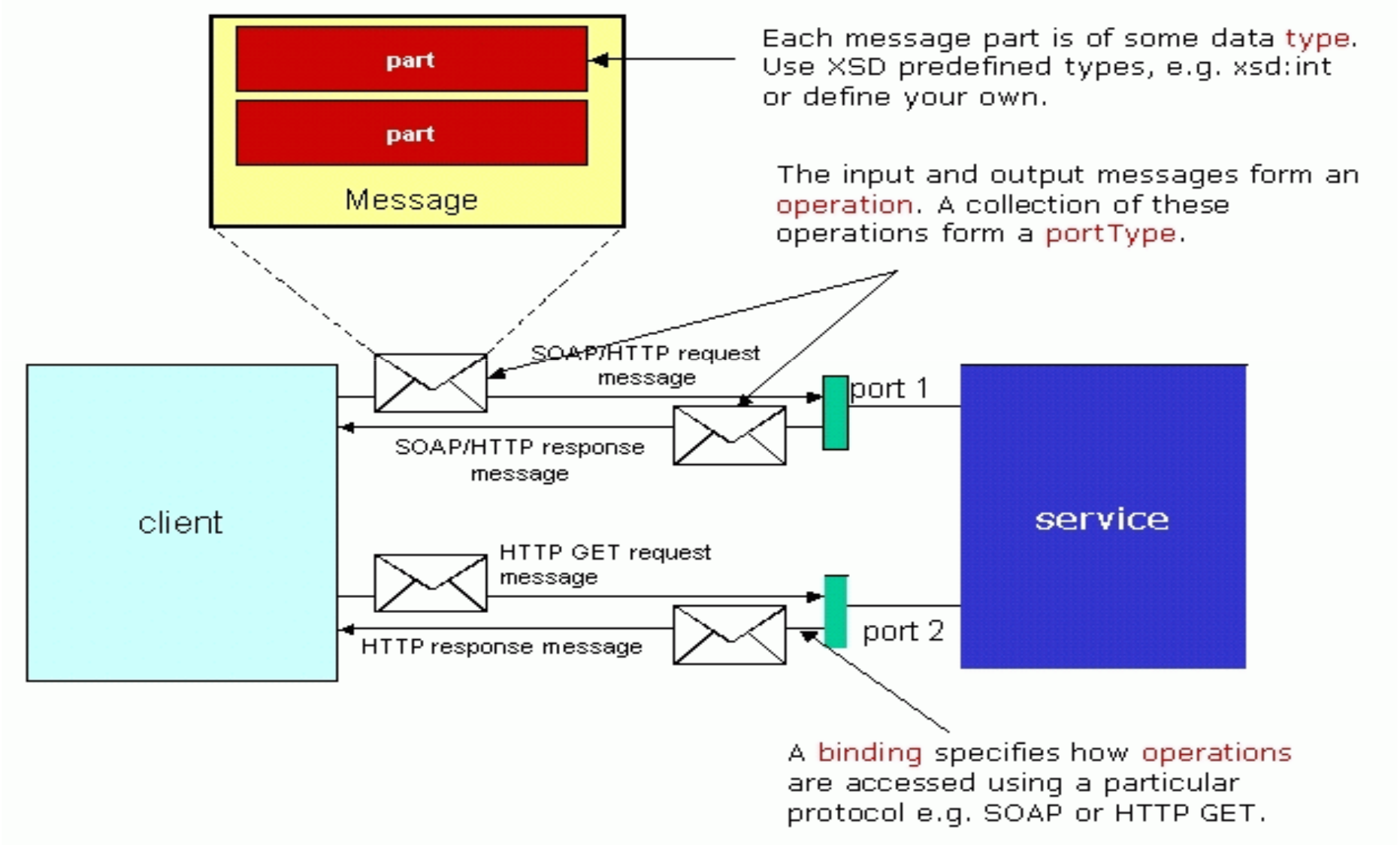
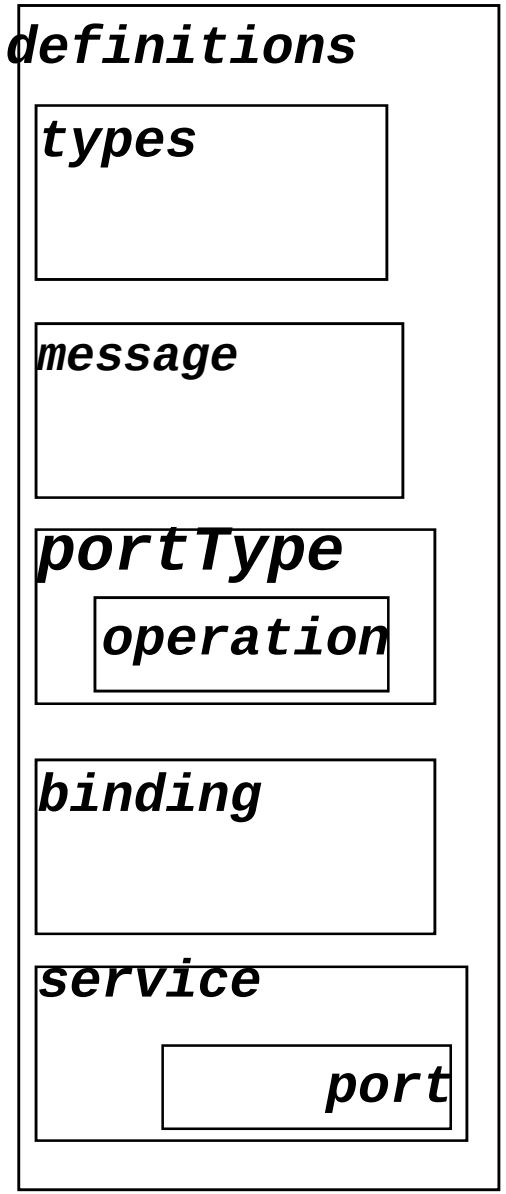


Figure 2. WSDL terminology used for describing Web services.

WSDL Components



All the data types used by the Web service
XML Schema
Namespace: XML namespaces

Parameters and messages used by method

Abstract interface definition – each *operation* element defines a method signature

Binds abstract methods to specific protocols
SOAP, HTTP, MIME and etc.

A service is a collection of ports.
A port is a specific method and its URI

The Main Structure of WSDL

<definition **namespace** = “http/… “>

<**type**> xschema types </type>

<**message**> … </message>

<**port**> a set of operations </port>

<**binding**> communication protocols </binding>

<**service**> a list of binding and ports </service>

<definition>

Namespace

- The XML namespace prefix are used to indicate the namespace of the element being defined
- All WSDL elements belong to the WSDL namespace, defined as http://schemas.xmlsoap.org/wsdl/
- For WSDL SOAP binding,
  http://schemas.xmlsoap.org/wsdl/soap/
- For WSDL HTTP GET and POST binding,
  http://schemas.xmlsoap.org/wsdl/http/
- For WSDL MIME binding,
  http://schemas.xmlsoap.org/wsdl/mime/

MIME: Multipurpose Internet Mail Extensions, a specification for formatting non-ASCII messages so that they can be sent over the Internet

Types

- <types> define types used in message declaration
- XML Schema, DTD, and etc.
- XML Schema must be supported by any vendor of WSDL conformant products.

```
<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“
                minOccur = “1”  maxOccur=“10”/>
              <element name = “payment”>
                 <complexType> <choice>
                    <element name = “account” type=“string”>
                    <element name = “creditcard” type=“string”>
                 </choice> </complexType>
               </element>
           </all>
        </complexType>
     </element>
  </schema>
</types>
```

Message

- A message is protocol independent
- There is an input or request message, which is sent from the client to the service, and there is a output or response message, which is sent back the opposite way
- Each <message> element contains one or more <part> elements.
- <part> element corresponds to the parameter or a return value in the RPC call.
- The part name order reflects the order of the parameters in the RPC signature.

Message contd.

```
<message name='Weather.GetTemperature' >
      <part name='zipcode' type='xsd:string' />
      <part name='celsius'  type='xsd:boolean' />
</message>
<message name='Weather.GetTemperatureResponse' >
      <part name='Result' type='xsd:float' />
</message>
```

WSDL Ports

- The **<portType>** element is the most important WSDL element.
- It defines **a web service**, the **operations** that can be performed, and the **messages** that are involved.
- The <**port**> defines the connection point to a web service, an instance of <**portType**>.
- It can be compared to a function library (or a module, or a class) in a traditional programming language. Each operation can be compared to a function in a traditional programming language.

WSDL Ports

- A port defines an individual endpoint by specifying a single address for a binding

```
<port name='WeatherSoapPort' binding='wsdlns:WeatherSoapBinding' >
        <soap:address
        location='http://localhost/demos/wsdl/devx
        pert/weatherservice.asp' />
    </port>
```

- Each port has a unique name and a binding attribute
- A web service may be accessible on many ports
- A port MUST NOT specify more than one address
- A port MUST NOT specify any binding information other than address information

Operations and PortType

- Operation defines which message is the input and which message is the output
- A collection of all operations exposed bythe web service is called a portType

```
<portType name='WeatherSoapPort'>
    <operation name='GetTemperature' parameterOrder='zipcode
    celsius'>
        <input message='wsdlns:Weather.GetTemperature' />
        <output message='wsdlns:Weather.GetTemperatureResponse'
    />
    </operation>
    <!– other operations →
</portType>
```

WSDL Ports

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

Operations and PortType

- WSDL has four transmission primitives that an endpoint can support:

➢ One-Way – The endpoint receives a message

<wsdl:input>

➢ Request-response – The endpoint receives a message and sends a correlated message

<wsdl:input>,<wsdl:output>,<wsdl:fault>

➢ Solicit-response – The endpoint sends a message and receives a correlated message

<wsdl:output>, <wsdl:input>, <wsdl:fault>

➢ Notification – The endpoint sends a message

<wsdl:output>

One way and Notification Example

```
<portType name=“RegisterPort">
   <operation name=“register">
      <input name=“customerInfo" message=“RegInfo"/>
   </operation>

   <operation name = “register Response”>
      <output name = “response” message=“ResponseInfo”/>
   </operation>
</portType >
```

Binding

- Binding mechanism is used to attach a specific protocol, data format or structure to an abstract message , operation or endpoint
- Binding MUST specify exactly one protocol
- Binding MUST NOT specify address information
- Extensibility elements are commonly used to specify some technology specific binding

```
<binding name='WeatherSoapBinding' type='wsdlns:WeatherSoapPort' >
    …
</binding>
```

SOAP Binding

- <soap:binding> - Signifies that the binding is bound to the SOAP protocol format: Envelope, Header and Body

```
<binding …>
    <soap:binding transport="uri"? Style="rpc|document"?>
</binding>
```

- <soap:operation> - Provides information for the document as a whole

```
<binding …>
    <operation …>
        <soap:operation soapAction="uri"?
    Style="rpc|document"?>
    </operation>
</binding>
```

SOAP Binding contd.

- <soap:body> - Specifies how the message parts appear inside the SOAP Body element

<input>

<soap:body parts="nmtokens"? use="literal|encoded"?

encodingStyle="uri-list"? Namespace="uri"?>

</input>

- <soap:fault> - Specifies the contents of the contents of the SOAP fault

<fault>

<soap:fault name="nmtoken" use="literal|encoded" encodingStyle="uri-list"? Namespace="uri"?>

</fault>

SOAP binding contd.

- <soap:header> and <soap:headerfault> - Allow headers to be defined that are transmitted inside the Header element of the SOAP Envelope

```
<input>
    <soap:header message="qname" part="nmtoken"
    use="literal|encoded"? encodingStyle="uri-list"?
    Namespace="uri"?>
    <soap:headerfault message="qname" part="nmtoken"
    use="literal|encoded"? encodingStyle="uri-list"?
    Namespace="uri"?>
</input>
```

- <soap:address> - Used to give a port an address (a URI)

```
<binding …>
    <soap:address location="uri" />
</binding>
```

HTTP GET & POST Binding

- <http:address> - Specifies the base URI for the port
- <http:binding> - Indicates that this binding uses the HTTP protocol

```
<binding …>
    <http:binding verb=“nmtoken” />
</binding>
```

- <http:operation> - has an attribute that specifies the relative URI for the operation

```
<operation …>
    <http:operation location=“uri” />
</operation>
```

HTTP GET & POST Binding contd.

- <http:urlEncoded> - Indicates that all the parts are encoded into the HTTP request URI using the standard URI-encoding rules
- <http:urlReplacement> - Indicates that all the message parts are encoded into the HTTP request URI using a replacement algorithm

MIME Binding

- <mime:content> - Used if there is no additional information to convey about the format other than its MIME type string

  <mime:content part="nmtoken"? Type="string"? />

- <mime:multipartRelated> - Aggregates an arbitrary set of MIME formatted parts into one message using the MIME type "multipart/related"

<mime:multipartRelated>

  <mime:part> *

    ← mime element →

  </mime:part>

</mime:multipartRelated>

MIME Binding contd.

- <soap:body> - When using the MIME binding with SOAP requests, it is legal to use the soap:body element as a MIME element. It indicates the content type is “text/xml”, and there is an enclosing SOAP Envelope
- <mime:mimeXml> - Used to specify a concrete schema

<mime:mimeXml part=“nmtoken”? />

```
<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>
```

```
<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
  </port>
</service>
```

Sample WSDL: getQuote

```
<?xml version="1.0" encoding="UTF-8" ?>

<definitions name="net.xmethods.services.stockquote.StockQuote"
targetNamespace="http://www.themindelectric.com/wsdl/net.xmethods.services.stockquote.StockQuote/"
xmlns:tns="http://www.themindelectric.com/wsdl/net.xmethods.services.stockquote.StockQuote/"
   xmlns:electric="http://www.themindelectric.com/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/">

<message name="getQuoteResponse1">
    <part name="Result" type="xsd:float" />
</message>

<message name="getQuoteRequest1">
    <part name="symbol" type="xsd:string" />
</message>
```

Sample WSDL: getQuote

```
<portType name="net.xmethods.services.stockquote.StockQuotePortType">
   <operation name="getQuote" parameterOrder="symbol">
     <input message="tns:getQuoteRequest1" />
     <output message="tns:getQuoteResponse1" />
   </operation>
</portType>

<binding name="net.xmethods.services.stockquote.StockQuoteBinding“
         type="tns:net.xmethods.services.stockquote.StockQuotePortType">
    <soap:binding style="rpc"
                  transport="http://schemas.xmlsoap.org/soap/http" />
    <operation name="getQuote">
       <soap:operation soapAction="urn:xmethods-delayed-quotes#getQuote" />
         <input>
           <soap:body use="encoded" namespace="urn:xmethods-delayed-quotes"
             encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
         </input>
         <output>
           <soap:body use="encoded" namespace="urn:xmethods-delayed-quotes"
             encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
         </output>
    </operation>
</binding>
```

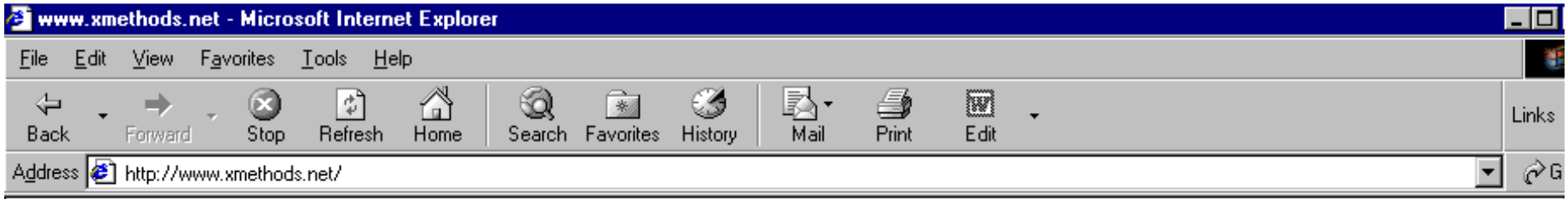
Sample WSDL: getQuote

```
<service name="net.xmethods.services.stockquote.StockQuoteService">
  <documentation>net.xmethods.services.stockquote.StockQuote web service
  </documentation>
   <port name="net.xmethods.services.stockquote.StockQuotePort"
         binding="tns:net.xmethods.services.stockquote.StockQuoteBinding">
     <soap:address location="http://64.39.29.211:9090/soap" />
   </port>
</service>

</definitions>
```

WSDL to Code

- Translators available that can
  - Convert WSDL document to code
    - IBM's WSTK Toolkit
    - Apache AXIS WSDL2java program
    - Soapy.py in Python
    - Not perfect
  - Derive WSDL from Java classes
    - Apache WSDL program
    - Much work remains to be done



XMETHODS

Services · Implementations · Manage · Register · Tutorials · Mailing List · About

Welcome to XMethods.

Emerging standards such as SOAP will enable a new generation of "web services" that allow systems to make remote procedure calls to other systems over the Internet. For example, a corporate inventory management sytem might publish a service that allows a customer system to check real-time inventory levels. This site lists publicly accessible web services.

Updates

2002-01-30	Configuring SOAP calls with the WSDL Analyzer [Read]
2002-01-15	SITE UPGRADED [Read]
2002-01-14	All WSDL on the site has been validated. [Read]
2001-01-14	Sign up to be notified of new services. [Read]

SOAP Services

Owner	Service Name	Description	Implementation
esynaps	eSynapsFeed	Daily Articles. Coding Tips and .NET Code samples	MS .NET
mybubble.com	Company Profile	Provides company profile for a given stock ticker	Apache SOAP
mybubble.com	Current News for a Stock	Provides current news of a company for a given stock ticker.	Apache SOAP
mybubble.com	Stock Quotes	Provides current quotes and additional info. for a given stock ticker.	Apache SOAP
myezconnect	Loan Term Worksheet	Loan Term Worksheet	GLUE
myezconnect	Extra Payment Worksheet	Extra Payment Worksheet	GLUE
myezconnect	Financial Calculator	Monthly Payment Worksheet	GLUE
geographynetwork.com	PlaceFinder	Returns the x,y location for a place name in any part of the world.	GLUE
daniel	Chess	Play Chess with a WebService	Delphi
esynaps	NFL Headline News	Get the NFL Headline News	MS .NET
esynaps	Who Is	The Web Service form of "WhoIs" Domain Registry service	MS .NET
esynaps	Daily Dilbert	Returns a binary stream of Today's Dilbert comic strip	MS .NET
eltegra.com	Monthly Mortgage Payment	Calculates your monthly mortgage payment	EXADEL
jcono	Location Information	Info about a location from zip code, area code, city, or state	MS .NET
caryjensen	Temperature Conversion Service	Converts Fahrenheit to Centigrade and vice versa	Delphi
abysal.com	SendEmail	Send a simple e-mail	Abysal-webDTP
simonfell	whois	A SOAP version of the standard whois service	4s4c
GIServer	GIServer Location Services	Country Location and Projections Transformation of Geographic Points using inovaGIS objects	Delphi

Problems

- Complex long running web services.
- Communication infrastructure –
  - Asynchronous communication
- Analysis and verification
- Automatic composition of complex services.

…