XML and Web Services

Ms. T.Anusha
Assistant Professor(Sel. Grade)
Department of Computer Science and Engineering
PSG College of Technology
Coimbatore

Simple Object Access Protocol

- Open Specification
- Simple to “write” and “human-readable”
- Extensible
- Flexible
- Standard way of serializing the information needed to invoke services located on remote system, in a format the remote system can understand.

Remote Procedure Call(RPC) architecture

- Internet Inter-ORB Protocol that underlies CORBA
- Microsoft’s Distributed Component Object Model (DCOM) protocol

Note: SOAP can be thought of as a simple XML-based replacement for these protocol

Improved RPC

Message Passing/Queuing

A message sender can send a message at any time, and the messaging infrastructure is responsible for delivering the message whenever it can, thus typically offering asynchronous message delivery.

Request/Response

With the request/response model, the message sender typically must wait until it receives a response from the recipient and it is an example for synchronous message delivery.

Problems faced by CORBA/IIOP and DCOM

- Both CORBA and DCOM are single-vendor solutions
- Both CORBA and DCOM have different proprietary characteristics
- IIOP and DCOM are both binary protocols
- Both CORBA and DCOM are tightly coupled

- SOAP is a vendor-neutral protocol
- SOAP interfaces are described with the Web Services Description Language(WSDL)
- SOAP is based on XML and it is a text-based protocol
- SOAP is a loosely coupled protocol

Improved Interoperability

- DCOM and CORBA are technically interoperable
- The custom integration make such bridges work in real-world environments is extraordinarily expensive and time consuming
- Integration: Translating Network Data Representation(NDR) payloads to Common Data Representation(CDR) payloads
- Sending executable code over the network opens a Pandora’s box of issues including security, flexibility and the previously mentioned firewall unfriendliness.

- SOAP is the XML-based replacement for the object serialization techniques used by the existing OPRC architectures.
- Using XML to structure the data serialization provides a “neutral third party” between CORBA and DCOM
- CORBA-SOAP and DCOM-SOAP bridges are much simpler to build and use than a CORBA-DCOM bridge, because they are simply XML interfaces to existing objects.

Key Building Block for Web Services

- SOAP messages are self-describing
- SOAP is ideally suited both for messages between Web Services and for messages that other systems exchange with Web Services

Basic SOAP Syntax

- SOAP is a messaging framework, consisting of an outer Envelope element that contains an optional Header element and a mandatory Body element.
- SOAP is an encoding format that describes how objects are encoded, serialized and then decoded when received.
- SOAP is an RPC mechanism that enables objects to call methods of remote objects.

SOAP Message Structure and Namespaces

```
public interface PhoneNumber
{
    public String getPhoneNumber(String name);
}
<?xml version=“1.0”?>
<PhoneNumber>
    <getPhoneNumber>
        <name>John Doe</name>
    </getPhoneNumber>
</PhoneNumber>
```

Response from the Server

```
<?xml version="1.0"?>
<PhoneNumber>
    <getPhoneNumberResponse>
        <thenumber>
            <areacode>617</areacode>
            <numberbody>555-1234</numberbody>
        </thenumber>
    </getPhoneNumberResponse>
</PhoneNumber>
```

SOAP Envelope Element

- Mandatory top element of the XML document that represents the SOAP message being sent.
- It may contain namespace declaration as well as other attributes which must be “namespace qualified”.
- The Envelope element may also contain additional sub elements which must also be namespace qualified and follow the Body element.

SOAP Header Element

The SOAP Header element is optional and is used for extending messages without any sort of prior agreement between the two communicating parties.

```
<SOAP.ENV:Header>
    <t:Transaction xmlns:t="myURI"
        SOAP.ENV:mustUnderstand="1">
    3
    </t:Transaction>
</SOAP.ENV:Header>
```

SOAP Body Element

- The mandatory Body element is an immediate child of the Envelope element and must immediately follow the Header element if a header is present.
- Each immediate child of the Body element is called a body entry.
- The Body element is used to carry the payload of the SOAP message.

Arrays

<SOAP.ENC: Array SOAP.ENC:arraytype=“xsd:int[2]”>

<SOAP.ENC:int>4<SOAP.ENC:int>

<SOAP.ENC:int>33</SOAP.ENC:int>

<SOAP.ENC:Array>

Structs

<elt: Purchase>
    <buyer>John Doe</buyer>
    <item>Widget</item>
    <count>2</count>
    <cost>14.47</cost>
</elt: Purchase>

Associated Schema Fragment

```
<element name=“Purchase”
    <complexType>
        <element name=“buyer” type=“xsd:string”/>
        <element name=“item” type=“xsd:string”/>
        <element name=“count” type=“xsd:int”/>
        <element name=“cost” type=“xsd:decimal”/>
    </complexType>
</element>
```

Thank
You