

SOAP and Its Extensions

Matt Van Gundy

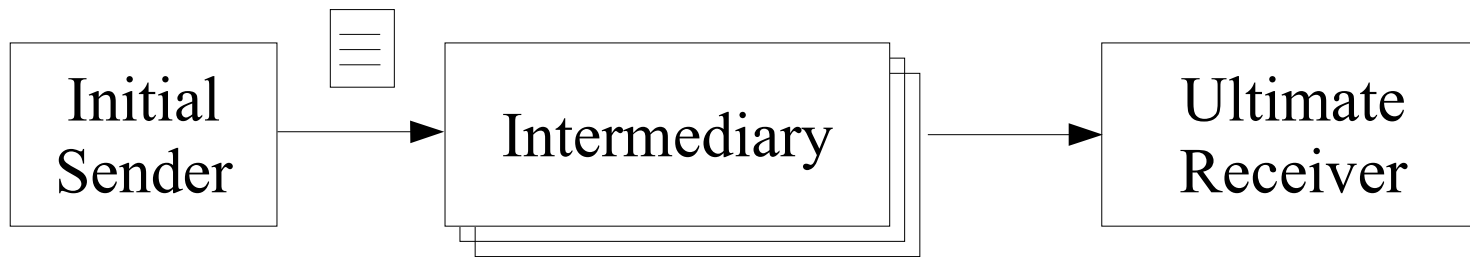
CS 595G

2006.02.07

What is SOAP?

- Formerly Simple Object Access Protocol
- Abstract Stateless Messaging Protocol
 - Another XML-based Meta-Standard
- SOAP Processing Model
- SOAP Message Construct
- SOAP Features
- SOAP Protocol Binding Framework

SOAP Processing Model & Roles



SOAP Roles (specified by URN):

- next
- none
- ultimateReceiver
- Application Specific

The SOAP Message Construct

- XML Infoset

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">  
  <soap:Header>  
    <!-- Optional: contents are application specific -->  
  </soap:Header>  
  <soap:Body>  
    <!-- Required: contents are application specific -->  
  </soap:Body>  
</soap:Envelope>
```

The SOAP Message Construct

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
  <soap:Header>
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol"
      soap:role=".../next" soap:mustUnderstand="false" soap:relay="true">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </soap:Header>
  <soap:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </soap:Body>
</soap:Envelope>
```

SOAP Attributes

- `encodingStyle:URI` – designates the serialization method used on element
- `mustUnderstand:boolean` – targeted intermediary must understand and process indicated header according to its specifications
- `relay:boolean` – indicates that a header is to be relayed if it is not understood by the intermediary assuming role the header is targeted to
- `role:URI` – designates the target of a header

SOAP Processing Model Rules

Upon receiving a message a node must:

1. Determine its role(s) (may use message contents)
2. Identify all mustUnderstand header blocks targeted at its role(s)
3. Generate a fault if any of the mustUnderstand header blocks are not understood
4. Process all headers targeted to the node and if the node is the ultimateReceiver process the body
5. If not the ultimateReceiver, relay the message

Example SOAP Request

```
POST /onca/soap?Service=AWSECommerceService HTTP/1.1
User-Agent: Mono Web Services Client Protocol 1.1.4322.2032
SOAPAction: "http://soap.amazon.com"
Content-Type: text/xml; charset=utf-8
Content-Length: 475
Expect: 100-continue
Connection: keep-alive
Host: soap.amazon.com
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ItemSearch xmlns="http://webservices.amazon.com/AWSECommerceService/2005-10-05">
      <AWSAccessKeyId>...</AWSAccessKeyId>
      <Shared>
        <Keywords>Led Zeppelin</Keywords>
        <SearchIndex>Music</SearchIndex>
      </Shared>
    </ItemSearch>
  </soap:Body>
</soap:Envelope>
```


Example SOAP Response

HTTP/1.1 200 OK

Date: Tue, 07 Feb 2006 00:00:52 GMT

Server: Server

nnCoection: close [sic]

Transfer-Encoding: chunked

Content-Type: text/xml; charset=UTF-8

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="...">
  <SOAP-ENV:Body>
    <ItemSearchResponse xmlns="http://.../AWSECommerceService/2005-10-05">
      <Items>
        <Request><IsValid>True</IsValid></Request>
        <TotalResults>301</TotalResults>
        <TotalPages>31</TotalPages>
        <Item>
          <ASIN>B000002J09</ASIN>
          <DetailPageURL>...</DetailPageURL>
          <ItemAttributes>
            <Artist>Led Zeppelin</Artist>
            <ProductGroup>Music</ProductGroup>
            <Title>Led Zeppelin IV (aka ZOSO)</Title>
          </ItemAttributes>
        </Item>
        ...
      </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>
```

SOAP Faults

...

```
<soap:Body>
  <soap:Fault>
    <soap:Code>
      <soap:Value>
        <!-- VersionMismatch | MustUnderstand | DataEncodingUnknown | Sender
          | Receiver | Application Specific -->
      </soap:Value>
      <soap:Subcode><!-- Optional -->
        <soap:Value>...</soap:Value>
        <soap:Subcode>...</soap:Subcode>
      </soap:Subcode>
    </soap:Code>
    <soap:Reason>
      <soap:Text><!-- Human readable message --></soap:Text>
      ...
    </soap:Reason>
    <soap:Node><!-- Optional: URI of faulting Node --></soap:Node>
    <soap:Role><!-- Optional: URI of role Node was playing --></soap:Role>
    <soap:Detail><!-- Optional: user specified XML --></soap:Detail>
  </soap:Fault>
</soap:Body>
```

SOAP Omissions

SOAP attempts to be extensible by 'omitting, from the messaging framework, features that are often found in distributed systems. Such features include but are not limited to "reliability", "security", "correlation", "routing", and "Message Exchange Patterns" (MEPs).'

SOAP Features

SOAP Features constrain an aspect of the Processing Model or Protocol Binding via:

- A URI used to name the feature
- The information (state) required at each node to implement the feature
- The processing required at each node in order to fulfill the obligations of the feature
- The information to be transmitted from node to node

SOAP Protocol Binding Framework

A SOAP binding specification:

- Declares the features provided by a binding.
- Describes how the services of the underlying protocol are used to transmit SOAP message infosets.
- Describes how the services of the underlying protocol are used to honor the contract formed by the features supported by that binding.
- Describes the handling of all potential failures that can be anticipated within the binding.
- Defines the requirements for building a conformant implementation of the binding being specified.

SOAP Adjuncts

- SOAP Encoding
- SOAP RPC Feature
- SOAP HTTP Protocol Binding

Points of Security Interest

- Processing Instruction Information Items
- 2048 Character URIs
- WS-Routing – possible source specified routing
- XML DSig c14n

