Using SOAP HTTP as an MTOSI Transport

Abstract

This document illustrates how MTOSI messages can be encapsulated in SOAP and transported over HTTP in support of the MTOSI application level requirements for communication and operation exchange.

It gives an introduction to the HTTP SOAP concepts and then it gives the bindings rules and recommendations to use SOAP encapsulated message over HTTP to support MTOSI operations.

Table of Contents

1 Introduction

MTOSI messages are defined in a way that is transport independent. Details of the communication concepts requested at the application level and the different styles supported by MTOSI are presented in [1]. OSs communicate through the CCV abstraction (Common Communication Vehicle).

A service consumer interacts with a service provider through the invocation of operations to achieve a business goal. The operations involve exchanges of XML document containing messages. The messages are exchanged using a particular communication style. In addition, a communication pattern identifies the sequence and cardinality of messages sent and/or received as well as whom they are sent to or received from.

Two communication styles are promoted by MTOSI, Messaging and RPC, and for each of them there are four communication patterns considered: `SimpleResponse`, `MultipleBatchResponse`, `BulkResponse` and `Notification`.

MTOSI supports both the Messaging communication style (`MSG`), and the RPC communication style (`RPC`). The RPC communication style (`RPC`) is best supported by transport technologies such as HTTP/S or CORBA. While JMS can also support RPC communication style, MTOSI recommends that JMS transport be used in conjunction with the MSG communication style as described in [3].

This document illustrates how SOAP encapsulated messages transported over HTTP (SOAP/HTTP) can be used to implement the CCV as a transport mechanism supporting the MTOSI application level requirements in terms of communication and operation exchange.

Throughout this document, we present different facets of SOAP/HTTP and show how they fit with the MTOSI requirements. Different examples, using code snippets, are used for illustrative purposes.

There are many reasons why SOAP/HTTP is appropriate for supporting the MTOSI RPC communication requirements. While this document focuses mainly on the technical ones, it is worth mentioning some others:

1. SOAP/HTTP is available in many middleware products
2. It is a W3C standard complete enough to avoid unnecessary proprietary extensions..

The document has the following main sections:

- Section 2 presents the essential concept of the SOAP/HTTP framework for readers not familiar with it. Experts in SOAP/HTTP may skip this section.

Section 3 presents MTOSI rules (denoted as **Rxx**)
and recommendations (denoted as **Oxx**) concerning how SOAP/HTTP should be used as a transport mechanism for MTOSI, and illustrates the different patterns through examples.

2 Overview of the HTTP / SOAP

SOAP V1.1 [3]~~[3]~~ was accepted as a W3C recommendation in May 2000, as proposed by UserLand, Ariba, Commerce One, Compaq, Developmentor, HP, IBM, IONA, Lotus, Microsoft, and SAP.

SOAP (Simple Object Access Protocol) is a simple XML-based protocol to facilitate the exchange of structured information between distributed applications in a decentralized environment.

SOAP can be used in combination with any kind of transport technology; in particular, MTOSI uses SOAP combined with JMS transport to support the message oriented communication style (see [2]~~[2]~~) and combined with HTTP transport for the RPC communication style.

The SOAP V1.1 document as described in [3]~~[3]~~ organizes the specification in several independent and complementary pieces:

- the format of a SOAP message
- how to serialize a message using the appropriate encoding rules
- conventions to represent RPC calls and responses
- how a SOAP message can be transported using the HTTP protocol with or without the HTTP Extension Protocol

While SOAP messages are used as one-way transmissions elements between a (unique) sender and a (unique) receiver, they are often used to implement patterns such as RPC request/response. Although SOAP for RPC is not limited to the HTTP protocol binding, an RPC call maps naturally to an HTTP request and an RPC response maps to an HTTP response.

The general format of the HTTP Header is shown in section 5~~5~~. Below, some specifics are summarized from the section describing the HTTP Binding in SOA V1.1 (section 6 of [3]~~[3]~~: “Using SOAP in HTTP”):

- HTTP applications MUST use the media type "text/xml" according to RFC 2376 when including SOAP entity bodies in HTTP messages.

- SOAP Request:

  - Although SOAP might be used in combination with a variety of HTTP request methods, [3]~~[3]~~ only defines SOAP within HTTP POST requests.

  - The SOAPAction HTTP header field is mandatory ~~for~~ when issuing a SOAP HTTP Request.

    The value of the SOAPAction determines what action the receiver needs to take. In reality the SOAPAction could be used for optimization to extract the discriminator without having to parse the SOAP payload (e.g. if "hello_world" operation is being invoked, the discriminator is "hello_world" which would be part of the SOAP message, however the SOAPAction would also have the value "hello_world" which could used as discriminator to dispatch).

    The presence and content of the SOAPAction header field can be used by servers such as firewalls to appropriately filter SOAP request messages in HTTP. The header field value of empty string ("") means that the intent of the SOAP message is provided by the HTTP Request-URI. No value means that there is no indication of the intent of the message.

- SOAP Response:
  - For SOAP Response, SOAP HTTP follows the semantics of the HTTP Status codes for communicating status information in HTTP. For example, a 2xx status code indicates that the client's request including the SOAP component was successfully received, understood, and accepted etc.

    In case of a SOAP error while processing the request, the SOAP HTTP server MUST issue an HTTP 500 "Internal Server Error" response and include a SOAP message in the response containing a SOAP Fault element indicating the SOAP processing error.

- SOAP Fault:

  The SOAP Fault element is used to carry error and/or status information within a SOAP message.

  The SOAP Fault element defines the following four subelements. Other Fault subelements MAY be present, provided they are namespace-qualified

faultcode

The faultcode element is intended for use by software to provide an algorithmic mechanism for identifying the fault. The faultcode MUST be present in a SOAP Fault element and the faultcode value MUST be a qualified. SOAP V1.1 defines a small set of SOAP fault codes covering basic SOAP faults.

faultstring

The faultstring element is intended to provide a human readable explanation of the fault and is not intended for algorithmic processing. The faultstring element is similar to the 'Reason-Phrase' defined by HTTP. It MUST be present in a SOAP Fault element and SHOULD provide at least some information explaining the nature of the fault.

faultactor

The faultactor element is intended to provide information about who caused the fault to happen within the message path. It is similar to the SOAP actor attribute, but instead of indicating the destination of the header entry, it indicates the source of the fault. The value of the faultactor attribute is a URI identifying the source. Applications that do not act as the ultimate destination of the SOAP message MUST include the faultactor element in a SOAP Fault element. The ultimate destination of a message MAY use the faultactor element to indicate explicitly that it generated the fault.

detail

The detail element is intended for carrying application specific error information related to the Body element. It MUST be present if the contents of the Body element could not be successfully processed. It MUST NOT be used to carry information about error information belonging to header entries. Detailed error information belonging to header entries MUST be carried within header entries.

The absence of the detail element in the Fault element indicates that the fault is not related to processing of the Body element. This can be used to distinguish whether the Body element was processed or not in case of a fault situation.

Example of a SOAP Request and Response in HTTP:

```
POST /StockQuote HTTP/1.1
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "http://electrocommerce.org/abc#MyMessage"

<SOAP:Envelope...>
…
</SOAP:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP:Envelope...>
…
</SOAP:Envelope>
```

3 Mapping MTOSI MEPS to WSDL SOAP HTTP Operations

3.1 Mapping to HTTP

As presented in [1], MTOSI supports four communication patterns in combination with two communication styles. RPC for synchronous communication and MSG for asynchronous communication.
Although HTTP transport can be used to support both communication styles, it is however much more “natural” to use HTTP for synchronous communication. For this reason, we discourage the usage of HTTP to map the MTOSI MEPs related to the asynchronous communication style (ARR, ABR, AFB, WSN asynchronous).

Below are some rules and options to follow when mapping MTOSI SOAP to HTTP:

R 1: MTOSI mandates the use of HTTP 1.1

O 1: HTTPS may be required by one of the communicating parties

R 2: MTOSI mandates the usage of SOAP V1.1
(SOAP V1.2 will be considered in future releases of MTOSI).

O 2: MTOSI does not mandate the usage of the HTTP Extension Protocol

R 3: A HTTP request message must use the HTTP POST method exclusively

R 4: As requested by SOAP V1.1, the media type to use must be “text/xml”
(“text” is the MIME media type name, and “xml” is the MIME subtype name)

R 5: MTOSI does not make use of the “mustUnderstand” attribute with any of the header entries of the SOAP headers specified in the “mtosiHeader” part. However, all those entries must be understood by the receiving party.

R 6: MTOSI requests that the HTTP POST *request URI* parameter is mapped from the *MTOSI activityName* (e.g. *getInventory*) of the SOAP header.

R 7: The value of the SOAPAction HTTP request header field is mapped from the *MTOSI msgName* (e.g. *getInventoryRequest*, *getInventoryIteratorRequest*) of the SOAP header

MTOSI uses SOAP/HTTP binding for RPC communication style with two patterns for **primitive operations**:

- request/response/fault

  In this pattern, the sender sends a SOAP request to a single receiver;
  the envelope of the SOAP request always contains a SOAP header and a SOAP body.
  If the receiver can process the request successfully it sends a single response back;
  the envelope of the SOAP response always contains a SOAP header and a SOAP body;
  If the receiver cannot process the request successfully, it sends a SOAP Fault message.

- notification

  In this pattern, the publisher (sender) sends a SOAP request (called "notify") to a single receiver;
  the envelope of the SOAP request always contains a SOAP header and a SOAP body.
  No response neither fault is expected from the receiver.

  The receiver can be the final receiver of the notification in case of Direct Notification or it can be the intermediate broker in case the Brokered Notification style is used [1]~~[1]~~ (in this latter case, the broker is in charge of forwarding the notification using as many "notify" SOAP requests as they are final receivers registered with the appropriate selector for the corresponding notification).

  In any case, this pattern means that the receiver of the "notify" SOAP request must behave as an HTTP server and the publisher must behave as an HTTP client.

The two figures below illustrate the two patterns using two examples of WSDL specification files:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- mTOP DDP - OM ManageResourceInventory - Copyright TeleManagement Forum 2007 -->
<wsdl:definitions name="mTOP-ResourceInventoryRetrievalV1-0" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:tns="http://www.tmforum.org/mtop/mri/wsdl/rir/v1-0"
targetNamespace="http://www.tmforum.org/mtop/mri/wsdl/rir/v1-0">
<wsdl:import namespace="http://www.tmforum.org/mtop/mri/wsdl/rir/v1-0" location="ResourceInventoryRetrievalMessages.wsdl"/>
 <wsdl:portType name="ResourceInventoryRetrieval_RPC">

   <wsdl:operation name="getInventory">
     <wsdl:input message="tns:getInventoryRequest"/>
     <wsdl:output message="tns:getInventoryResponse"/>
     <wsdl:fault name="getInventoryException" message="tns:getInventoryException"/>
   </wsdl:operation>

   <wsdl:operation name="getInventoryIterator">
             …
   </wsdl:operation>

 </wsdl:portType>

 <wsdl:binding name="ResourceInventoryRetrievalSoapHttpBinding" type="tns:ResourceInventoryRetrieval_RPC">

   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
   <wsdl:operation name="getInventory">
     <soap:operation soapAction="getInventory" style="document"/>
     <wsdl:input>
       <soap:header message="tns:getInventoryRequest" part="mtopHeader" use="literal"/>
       <soap:body parts="mtopBody" use="literal"/>
     </wsdl:input>
     <wsdl:output>
       <soap:header message="tns:getInventoryResponse" part="mtopHeader" use="literal"/>
       <soap:body parts="mtopBody" use="literal"/>
     </wsdl:output>
     <wsdl:fault name="getInventoryException">
       <soap:fault name="getInventoryException" use="literal"/>
     </wsdl:fault>
   </wsdl:operation>

   <wsdl:operation name="getInventoryIterator">
             …
   </wsdl:operation>

 </wsdl:binding>

 <wsdl:service name="ResourceInventoryRetrievalHttp">
   <wsdl:port name="ResourceInventoryRetrievalSoapHttp"             binding="tns:ResourceInventoryRetrievalSoapHttpBinding">
     <soap:address location="http://aserver/mtosi/ResourceInventoryRetrieval"/>
   </wsdl:port>
 </wsdl:service>
</wsdl:definitions>
```

Figure 1.The Request/Response/Fault pattern
(from file "ResourceInventoryRetrievalHttp.wsdl" – MRI DDP)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- mTOP DDP - Framework - Copyright TeleManagement Forum 2007 -->
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.tmforum.org/mtop/fmw/wsdl/notb/v1-0"
name="mTOP-NotificationBrokerV1-0" targetNamespace="http://www.tmforum.org/mtop/fmw/wsdl/notb/v1-0">
<wsdl:import namespace="http://www.tmforum.org/mtop/fmw/wsdl/notb/v1-0"
location="NotificationBrokerPortType.wsdl"/>

 <wsdl:binding name="NotificationBrokerSoapHttpBinding" type="tns:NotificationBroker">
   <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

   <wsdl:operation name="notify">
     <soap:operation soapAction="notify" style="document"/>
     <wsdl:input>
       <soap:header message="tns:notify" part="mtopHeader" use="literal"/>
       <soap:body parts="mtopBody" use="literal"/>
     </wsdl:input>
   </wsdl:operation>

   <wsdl:operation name="subscribe">
     <soap:operation soapAction="subscribe" style="document"/>
     <wsdl:input>
       <soap:header message="tns:subscribeRequest" part="mtopHeader" use="literal"/>
       <soap:body parts="mtopBody" use="literal"/>
     </wsdl:input>
     <wsdl:output>
       <soap:header message="tns:subscribeResponse" part="mtopHeader" use="literal"/>
       <soap:body parts="mtopBody" use="literal"/>
     </wsdl:output>
     <wsdl:fault name="subscribeException">
       <soap:fault name="subscribeException" use="literal"/>
     </wsdl:fault>
   </wsdl:operation>

   <wsdl:operation name="unsubscribe">
              …
   </wsdl:operation>
 </wsdl:binding>

 <wsdl:service name="NotificationBrokerHttp">
   <wsdl:port name="NotificationBrokerSoapHttp" binding="tns:NotificationBrokerSoapHttpBinding">
     <soap:address location="http://aserver/mtosi/NotificationBrokerV1-0"/>
   </wsdl:port>
 </wsdl:service>
</wsdl:definitions>
```

Figure 2. The Notification pattern
(from file "NotificationBrokerHttp.wsdl" – FMW DDP)

3.2 SimpleResponse MTOSI Communication Pattern

The ***simple response pattern*** involves a request/reply with a single result message. This pattern can be used with both the RPC or MSG communication styles:

- when used with the RPC communication style, the corresponding MEP is called SRR (Synchronous Request Reply)
- when used with the MSG communication pattern, the corresponding MEP is called ARR (Asynchronous Request Reply)

While it is indeed possible to emulate asynchronous communication over HTTP to support the ARR MEP, we do not encourage this practise.

On the opposite, the SRR MEP maps naturally to the HTTP transport. The figure below illustrates the dialogue sequence at the business level.

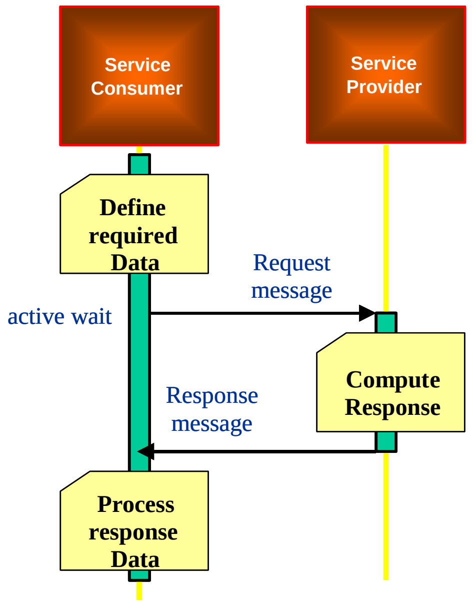


Figure 3. Dialogue sequence for the SRR MEP

To support this SRR MEP, the MTOSI header should include the mandatory fields and associated values as follows:

- activityName
- msgName
- msgType
- senderURI
- destinationURI
- activityStatus (response only)
- communicationPattern = SimpleResponse
- communicationStyle = RPC

The snippets below illustrate a SRR MEP request and its HTTP binding and a response for the getManagedElement business activity:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1"
xmlns:mer="http://www.tmforum.org/mtop/mri/xsd/mer/v1"
xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/mer/v1
../xsd/ManagedElementRetrievalMessages.xsd http://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
   <hdr:header>
     <hdr:activityName>getManagedElement</hdr:activityName>
     <hdr:msgName>getManagedElementRequest</hdr:msgName>
     <hdr:msgType>REQUEST</hdr:msgType>
     <hdr:senderURI>/MTOSI/InventoryOS</hdr:senderURI>
     <hdr:destinationURI>/MTOSI/EmsOS</hdr:destinationURI>
     <hdr:communicationPattern>SimpleResponse</hdr:communicationPattern>
     <hdr:communicationStyle>RPC</hdr:communicationStyle>
     <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
   </hdr:header>
 </soap:Header>
 <soap:Body>
   <mer:getManagedElementRequest>
     <mer:meName>
       <nam:rdn>
         <nam:type>MD</nam:type>
         <nam:value>string</nam:value>
       </nam:rdn>
       <nam:rdn>
         <nam:type>ME</nam:type>
         <nam:value>string</nam:value>
       </nam:rdn>
     </mer:meName>
   </mer:getManagedElementRequest>
 </soap:Body>
</soap:Envelope>
```

Figure 4. A getManagedElement request using the SRR MEP
(File “5-1_srr_getManagedElementRequest.xml” – MRI DDP)

```
POST getTP HTTP/1.1
Host: /MTOSI/EMS01
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
SOAPAction: "getTP"
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1" xmlns:mer="http://www.tmforum.org/mtop/mri/xsd/mer/v1"
xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/mer/v1 ../xsd/ManagedElementRetrievalMessages.xsd
http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
  <hdr:header>
   <hdr:activityName>getManagedElement</hdr:activityName>
   <hdr:msgName>getManagedElementRequest</hdr:msgName>
   <hdr:msgType>REQUEST</hdr:msgType>
   <hdr:senderURI>/MTOSI/InventoryOS</hdr:senderURI>
   <hdr:destinationURI>/MTOSI/EmsOS</hdr:destinationURI>
   <hdr:communicationPattern>SimpleResponse</hdr:communicationPattern>
   <hdr:communicationStyle>RPC</hdr:communicationStyle>
   <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
  </hdr:header>
 </soap:Header>
 <soap:Body>
  <mer:getManagedElementRequest>
   <mer:meName>
    <nam:rdn>
     <nam:type>MD</nam:type>
     <nam:value>string</nam:value>
    </nam:rdn>
    <nam:rdn>
     <nam:type>ME</nam:type>
     <nam:value>string</nam:value>
    </nam:rdn>
   </mer:meName>
  </mer:getManagedElementRequest>
 </soap:Body>
</soap:Envelope>
```

Figure 5. The getManagedElement request mapped to HTTP

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1"
xmlns:mer="http://www.tmforum.org/mtop/mri/xsd/mer/v1"
xmlns:coi="http://www.tmforum.org/mtop/fmw/xsd/coi/v1"
xmlns:gen="http://www.tmforum.org/mtop/fmw/xsd/gen/v1"
xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1"
xmlns:cri="http://www.tmforum.org/mtop/nrb/xsd/cri/v1" xmlns:lay="http://www.tmforum.org/mtop/nrb/xsd/lay/v1"
xmlns:lp="http://www.tmforum.org/mtop/nrb/xsd/lp/v1" xmlns:me="http://www.tmforum.org/mtop/nrf/xsd/me/v1"
xmlns:nt1="vendor.v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/mer/v1 ../xsd/ManagedElementRetrievalMessages.xsd
http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
  <hdr:header>
   <hdr:activityName>getManagedElement</hdr:activityName>
   <hdr:msgName>getManagedElementResponse</hdr:msgName>
   <hdr:msgType>RESPONSE</hdr:msgType>
   <hdr:senderURI>/MTOSI/EmsOS</hdr:senderURI>
   <hdr:destinationURI>/MTOSI/InventoryOS</hdr:destinationURI>
   <hdr:communicationPattern>SimpleResponse</hdr:communicationPattern>
   <hdr:communicationStyle>RPC</hdr:communicationStyle>
   <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
  </hdr:header>
 </soap:Header>
 <soap:Body>
  <mer:getManagedElementResponse>
   <mer:me>
    <coi:name>
     <nam:rdn>
      <nam:type>MD</nam:type>
      <nam:value>string</nam:value>
     </nam:rdn>
     <nam:rdn>
      <nam:type>ME</nam:type>
      <nam:value>string</nam:value>
     </nam:rdn>
    </coi:name>
    <coi:userLabel>string</coi:userLabel>
    <coi:discoveredName>string</coi:discoveredName>
    <coi:namingOs>string</coi:namingOs>
    <coi:owner>string</coi:owner>
    <coi:aliasNameList>
     <gen:alias>
      <gen:aliasName>string</gen:aliasName>
      <gen:aliasValue>string</gen:aliasValue>
     </gen:alias>
    </coi:aliasNameList>
    <cri:source>NETWORK_EMS</cri:source>
    <cri:networkAccessDomain>string</cri:networkAccessDomain>
    <cri:resourceState>INSTALLING</cri:resourceState>
    <me:location>string</me:location>
    <me:manufacturer>string</me:manufacturer>
    <me:productName>string</me:productName>
    <me:softwareVersion>string</me:softwareVersion>
    <me:isInSyncState>true</me:isInSyncState>
    <me:supportedConnectionLayerRateList>
     <lay:layerRate>LR_Line_OC48_STS48_and_MS_STM16</lay:layerRate>
     <lay:layerRate>LR_Line_OC192_STS192_and_MS_STM64</lay:layerRate>
```

```
        </me:supportedConnectionLayerRateList>
        <me:communicationState>CS_AVAILABLE</me:communicationState>
      </mer:me>
    </mer:getManagedElementResponse>
  </soap:Body>
</soap:Envelope>
```

Figure 6. A getManagedElement response using the SRR MEP
(File “5-2_srr_getManagedElementResponse.xml” – MRI DDP)

3.3 MultipleBatchResponse MTOSI Communication Pattern

The ***multiple batch response pattern*** is used when the request is expected to generate a large amount of resulting data to be returned. The response data is fragmented to several logical units.

This pattern can be used with both the RPC or MSG communication styles:

- when used with the RPC communication style, the corresponding MEP is called SIT (Synchronous Iterator)
- when used with the MSG communication pattern, the corresponding MEP is called ABR (Asynchronous Batch Response)

While it is indeed possible to emulate asynchronous communication over HTTP to support the ABR MEP, MTOSI do not encourage this practise.

The figure below, extracted from [1] illustrates the dialogue sequence at the business level when the SIT MEP is used.

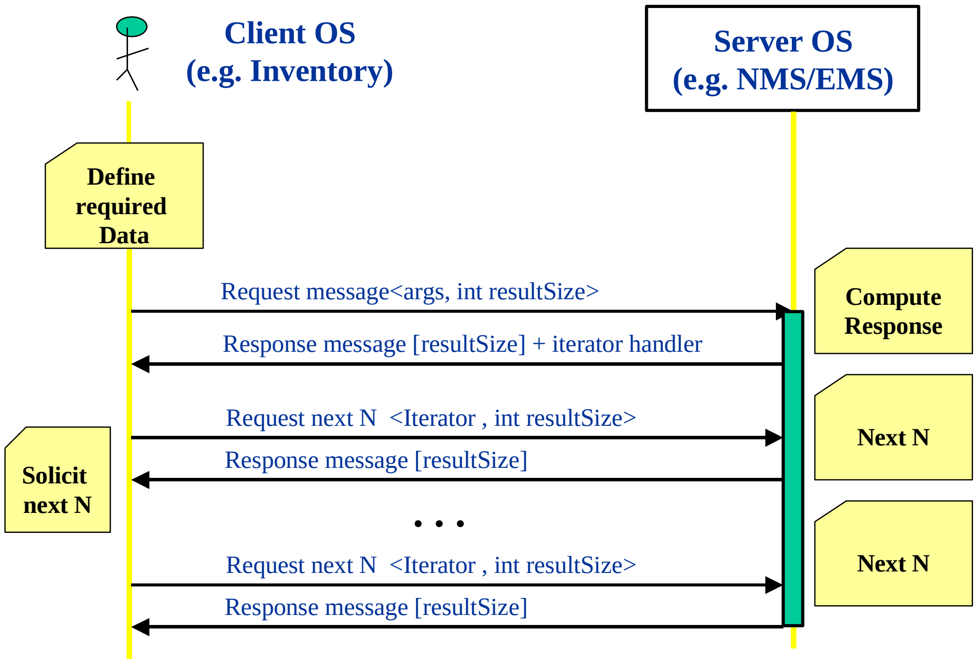

Client OS
(e.g. Inventory)
Server OS
(e.g. NMS/EMS)
Define required Data
Compute Response
Request message<args, int resultSize>
Response message [resultSize] + iterator handler
Request next N <Iterator , int resultSize>
Response message [resultSize]
Solicit next N
Next N
. . .
Request next N <Iterator , int resultSize>
Response message [resultSize]
Next N

Figure 7. Dialogue sequence for the SIT MEP

To support this SIT MEP, the MTOSI header should include the mandatory fields and associated values as follows:

Mandatory MEP elements in both request and responses:

- activityName
- msgName
- msgType
- senderURI
- destinationURI
- communicationPattern = MultipleBatchResponse

- communicationStyle = RPC

Mandatory MEP elements in requests (both initial request and Iterator requests):

- requestedBatchSize
  logical size of the batch for the result fragments. For example in the getInventory it is the maximum number of Inventory top level objects to be included in an inventory XML file (batch). See TMF 517 for details on how the result set is partitioned.
  If set to 0 (zero) will imply a single response.

Mandatory MEP elements in responses (both initial response and Iterator responses):

- iteratorReferenceURI
  contains the Iterator endpoint reference
- batchSequenceNumber
  identifies the batch sequence number in a sequence.
- batchSequenceEndOfReply
  indicates that the corresponding partial response of the last of the sequence.
- activityStatus
  indicates (values: SUCCESS. FAILURE, WARNING) the result of the activity.

The following figure and SOAP messages illustrate the SIT MEP based on the getInventory business activity.

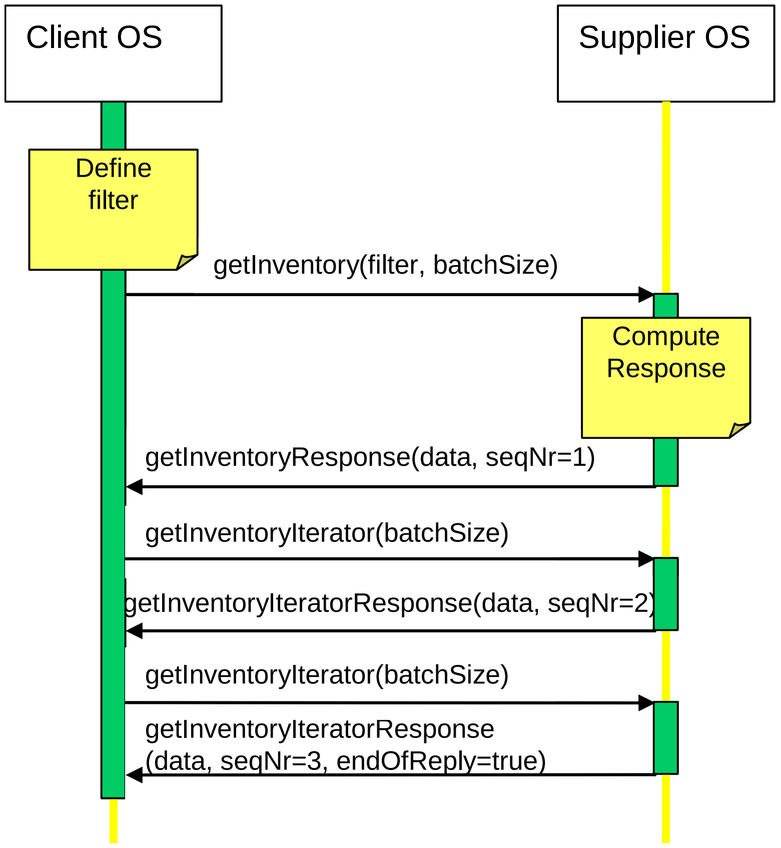

Client OS
Supplier OS
Define filter
getInventory(filter, batchSize)
Compute Response
getInventoryResponse(data, seqNr=1)
getInventoryIterator(batchSize)
getInventoryIteratorResponse(data, seqNr=2)
getInventoryIterator(batchSize)
getInventoryIteratorResponse (data, seqNr=3, endOfReply=true)

Figure 8. SIT MEP: dialogue sequence for the getInventory business activity

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

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1" xmlns:invr="http://www.tmforum.org/mtop/mri/xsd/rir/v1"
xmlns:gen="http://www.tmforum.org/mtop/fmw/xsd/gen/v1" xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/rir/v1
../xsd/ResourceInventoryRetrievalMessages.xsd  ://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
   <hdr:header>
     <hdr:activityName>getInventory</hdr:activityName>
     <hdr:msgName>getInventoryRequest</hdr:msgName>
     <hdr:msgType>REQUEST</hdr:msgType>
     <hdr:senderURI>/MTOSI/InventoryOS</hdr:senderURI>
     <hdr:destinationURI>/MTOSI/EMS01</hdr:destinationURI>
     <hdr:communicationPattern>MultipleBatchResponse</hdr:communicationPattern>
     <hdr:communicationStyle>RPC</hdr:communicationStyle>
     <hdr:requestedBatchSize>500</hdr:requestedBatchSize>
     <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
   </hdr:header>
 </soap:Header>
 <soap:Body>
   <invr:getInventoryRequest>
     <invr:filter>
       <invr:baseInstance>
         <nam:rdn>
           <nam:type>MD</nam:type>
           <nam:value>string</nam:value>
         </nam:rdn>
       </invr:baseInstance>
       <invr:includedObjectType>
         <invr:objectType>ME</invr:objectType>
         <invr:granularity>ATTRS</invr:granularity>
       </invr:includedObjectType>
       <invr:includedObjectType>
         <invr:objectType>PTP</invr:objectType>
          <invr:granularity>ATTRS</invr:granularity>
       </invr:includedObjectType>
       <invr:includedObjectType>
         <invr:objectType>CTP</invr:objectType>
         <invr:granularity>ATTRS</invr:granularity>
       </invr:includedObjectType>
     </invr:filter>
   </invr:getInventoryRequest>
 </soap:Body>
</soap:Envelope>
```

Figure 9. SIT MEP: initial getInventory request
(File “7-1_sit_getInventoryRequest.xml” – MRI DDP)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1" xmlns:invr="http://www.tmforum.org/mtop/mri/xsd/rir/v1"
xmlns:inv="http://www.tmforum.org/mtop/nrf/xsd/invdata/v1" xmlns:coi="http://www.tmforum.org/mtop/fmw/xsd/coi/v1"
xmlns:gen="http://www.tmforum.org/mtop/fmw/xsd/gen/v1" xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1"
xmlns:cri="http://www.tmforum.org/mtop/nrb/xsd/cri/v1" xmlns:lay="http://www.tmforum.org/mtop/nrb/xsd/lay/v1"
xmlns:lp="http://www.tmforum.org/mtop/nrb/xsd/lp/v1" xmlns:me="http://www.tmforum.org/mtop/nrf/xsd/me/v1"
xmlns:ptp="http://www.tmforum.org/mtop/nrf/xsd/ptp/v1" xmlns:ctp="http://www.tmforum.org/mtop/nrf/xsd/ctp/v1"
xmlns:nt1="vendor.v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/rir/v1 ../xsd/ResourceInventoryRetrievalMessages.xsd
://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
   <hdr:header>
     <hdr:activityName>getInventory</hdr:activityName>
     <hdr:msgName>getInventoryResponse</hdr:msgName>
     <hdr:msgType>RESPONSE</hdr:msgType>
     <hdr:senderURI>/MTOSI/EMS01</hdr:senderURI>
     <hdr:destinationURI>/MTOSI/InventoryOS</hdr:destinationURI>
     <hdr:communicationPattern>MultipleBatchResponse</hdr:communicationPattern>
     <hdr:communicationStyle>RPC</hdr:communicationStyle>
     <hdr:requestedBatchSize>500</hdr:requestedBatchSize>
     <hdr:batchSequenceNumber>1</hdr:batchSequenceNumber>
     <hdr:batchSequenceEndOfReply>false</hdr:batchSequenceEndOfReply>
     <hdr:iteratorReferenceURI>http://host:port/ResourceInventoryRetrievalV1/iterator123</hdr:iteratorReferenceURI>
     <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
   </hdr:header>
 </soap:Header>
 <soap:Body>
   <invr:getInventoryResponse>
     <invr:inventoryData>
       <inv:mdList>
         <inv:md>
           <inv:mdNm>MD_A</inv:mdNm>

           <inv:meList>
                                               ...
           </inv:meList>

     </invr:inventoryData>
   </invr:getInventoryResponse>
 </soap:Body>
</soap:Envelope>
```

Figure 10. SIT MEP: first getInventoryResponse
(File “7-2_sit_getInventoryResponse.xml” – MRI DDP)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1" xmlns:invr="http://www.tmforum.org/mtop/mri/xsd/rir/v1"
xmlns:gen="http://www.tmforum.org/mtop/fmw/xsd/gen/v1" xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/rir/v1
../xsd/ResourceInventoryRetrievalMessages.xsd  ://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
   <hdr:header>
     <hdr:activityName>getInventory</hdr:activityName>
     <hdr:msgName>getInventoryIteratorRequest</hdr:msgName>
     <hdr:msgType>REQUEST</hdr:msgType>
     <hdr:senderURI>/MTOSI/InventoryOS</hdr:senderURI>
     <hdr:destinationURI>/MTOSI/EMS01</hdr:destinationURI>
     <hdr:communicationPattern>MultipleBatchResponse</hdr:communicationPattern>
     <hdr:communicationStyle>RPC</hdr:communicationStyle>
     <hdr:iteratorReferenceURI>http://host:port/ResourceInventoryRetrievalV1/iterator123</hdr:iteratorReferenceURI>
     <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
   </hdr:header>
 </soap:Header>
 <soap:Body>
   <invr:getInventoryIteratorRequest/>
 </soap:Body>
</soap:Envelope>
```

Figure 11. SIT MEP: getInventoryIterator request
(File “7-3_sit_getInventoryIteratorRequest.xml” – MRI DDP)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1" xmlns:invr="http://www.tmforum.org/mtop/mri/xsd/rir/v1"
xmlns:inv="http://www.tmforum.org/mtop/nrf/xsd/invdata/v1" xmlns:coi="http://www.tmforum.org/mtop/fmw/xsd/coi/v1"
xmlns:gen="http://www.tmforum.org/mtop/fmw/xsd/gen/v1" xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1"
xmlns:cri="http://www.tmforum.org/mtop/nrb/xsd/cri/v1" xmlns:lay="http://www.tmforum.org/mtop/nrb/xsd/lay/v1"
xmlns:lp="http://www.tmforum.org/mtop/nrb/xsd/lp/v1" xmlns:me="http://www.tmforum.org/mtop/nrf/xsd/me/v1"
xmlns:ptp="http://www.tmforum.org/mtop/nrf/xsd/ptp/v1" xmlns:ctp="http://www.tmforum.org/mtop/nrf/xsd/ctp/v1"
xmlns:nt1="vendor.v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/rir/v1 ../xsd/ResourceInventoryRetrievalMessages.xsd
://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
   <hdr:header>
     <hdr:activityName>getInventory</hdr:activityName>
     <hdr:msgName>getInventoryIteratorResponse</hdr:msgName>
     <hdr:msgType>RESPONSE</hdr:msgType>
     <hdr:senderURI>/MTOSI/EMS01</hdr:senderURI>
     <hdr:destinationURI>/MTOSI/InventoryOS</hdr:destinationURI>
     <hdr:communicationPattern>MultipleBatchResponse</hdr:communicationPattern>
     <hdr:communicationStyle>RPC</hdr:communicationStyle>
     <hdr:requestedBatchSize>500</hdr:requestedBatchSize>
     <hdr:batchSequenceNumber>2</hdr:batchSequenceNumber>
     <hdr:batchSequenceEndOfReply>true</hdr:batchSequenceEndOfReply>
     <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
   </hdr:header>
 </soap:Header>
 <soap:Body>
   <invr:getInventoryIteratorResponse>
     <invr:inventoryData>
       <inv:mdList>
         <inv:md>
           <inv:mdNm>MD_A</inv:mdNm>

           <inv:meList>
                                        ...
           </inv:meList>

     </invr:inventoryData>
   </invr:getInventoryIteratorResponse>
 </soap:Body>
</soap:Envelope>
```

Figure 12. SIT MEP: final getInventoryResponse
(File “7-4_sit_getInventoryIteratorResponse.xml” – MRI DDP)

3.4 Synchronous File Bulk Response MTOSI Communication Pattern

The Bulk Response Pattern pattern is allows transferring the entire response set through a side channel (secondary to the main CCV transport) using a FTP server for instance.

This pattern can be used with both the RPC or MSG communication styles:

- when used with the RPC communication style, the corresponding MEP is called SFB (Synchronous File Bulk response)
- when used with the MSG communication pattern, the corresponding MEP is called AFB (Asynchronous File Bulk response)

The AFB MEP is the one that is more appropriate to be used over HTTP.

The figure below, extracted from [1] illustrates the dialogue sequence at the business level when the SFB MEP is used.

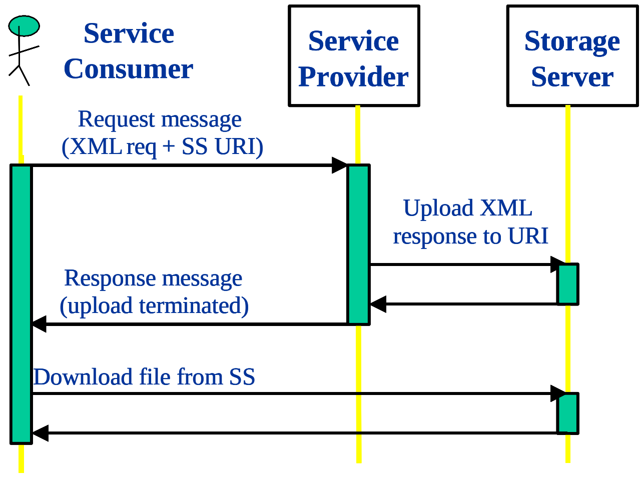

Service Consumer
Service Provider
Storage Server
Request message (XML req + SS URI)
Upload XML response to URI
Response message (upload terminated)
Download file from SS

Figure 13. Dialogue sequence for the SFB MEP

Mandatory MEP elements in both request and responses:

- activityName
- msgName
- msgType
- senderURI
- destinationURI
- communicationPattern = BulkResponse
- communicationStyle = RPC

Mandatory MEP elements in request:

- **requestedBatchSize**
  logical size of the batch for the transfer fragments. For example in the getInventory it is the maximum number of Inventory top level objects to be included in an inventory XML file (batch). See TMF 517 for details on how the result ret is partitioned.
  If set to 0 (zero) will imply a single response.

- **fileLocationURI**
  URI provided by the requesting OS indicating the rootname of the file(s) to be produced and location of where to place the retrieved XML inventory file(s). See RFC 3986 for details on URI syntax.
- **compressionType**
  The type of compression to apply to the generated file(s). MTOSI v1.1 will define the following two options; NO_COMPRESSION, GZIP. Default behavior (when request parameter is omitted) is NO_COMPRESSION. Implementation of this file processing instruction by the Target OS is optional, and any incompatible request shall be handled with the appropriate exception. Vendor extension of this attribute shall be permitted.
- **packingType**
  The type of packing to apply to all the inventory file(s) generated from the same request. MTOSI v1.1 will define the following three options; NO_PACKING, ZIP, TAR. Default behavior (when request parameter is omitted) is NO_ PACKING. Implementation of this file processing instruction by the Target OS is optional, and any incompatible request shall be handled with the appropriate exception. Vendor extension of this attribute shall be permitted.

Mandatory MEP elements in response:

The XML response to an AFB request is an acknowledgement of the FTP transfer and optionally a set of intermediate notification regarding the state of the FTP transfer (see use case in TMF 517).

The header element **activityStatus** should be used (values: SUCCESS. FAILURE, WARNING) in the acknowledgement to notify the service consumer of the result of the activity.

No additional elements are necessary in the header.

The following figure and SOAP messages illustrate the SFB MEP based on the getInventory business activity.

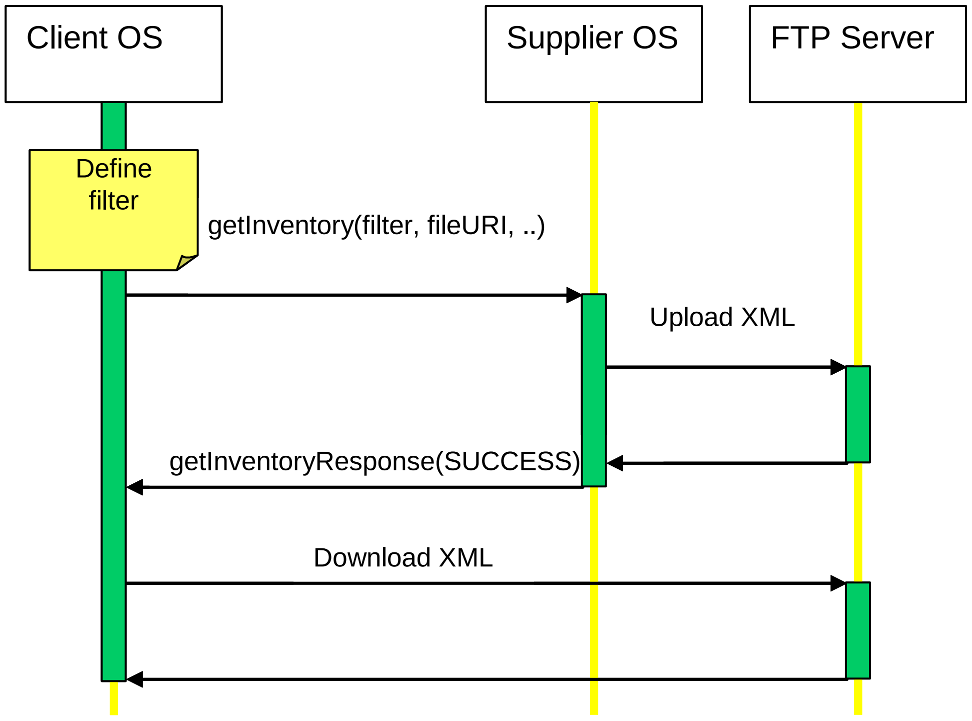


Figure 14. SFB MEP: dialogue sequence for the getInventory business activity

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1" xmlns:invr="http://www.tmforum.org/mtop/mri/xsd/rir/v1"
xmlns:gen="http://www.tmforum.org/mtop/fmw/xsd/gen/v1"
xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/rir/v1 ../xsd/ResourceInventoryRetrievalMessages.xsd
://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
   <hdr:header>
     <hdr:activityName>getInventory</hdr:activityName>
     <hdr:msgName>getInventoryRequest</hdr:msgName>
     <hdr:msgType>REQUEST</hdr:msgType>
     <hdr:senderURI>/MTOSI/InventoryOS</hdr:senderURI>
     <hdr:destinationURI>/MTOSI/EMS01</hdr:destinationURI>
     <hdr:communicationPattern>BulkResponse</hdr:communicationPattern>
     <hdr:communicationStyle>RPC</hdr:communicationStyle>
     <hdr:requestedBatchSize>500</hdr:requestedBatchSize>
     <hdr:fileLocationURI>ftp://user;password@host:port/path/filename</hdr:fileLocationURI>
     <hdr:compressionType>GZIP</hdr:compressionType>
     <hdr:packingType>ZIP</hdr:packingType>
     <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
   </hdr:header>
 </soap:Header>
 <soap:Body>
   <invr:getInventoryRequest>
     <invr:filter>
     <!-- Retrieve All ME, PTP and CTP object attributes in West MD -->
       <invr:baseInstance>
         <nam:rdn>
           <nam:type>MD</nam:type>
           <nam:value>West</nam:value>
         </nam:rdn>
       </invr:baseInstance>
       <invr:includedObjectType>
         <invr:objectType>ME</invr:objectType>
         <invr:granularity>ATTRS</invr:granularity>
       </invr:includedObjectType>
       <invr:includedObjectType>
         <invr:objectType>PTP</invr:objectType>
          <invr:granularity>ATTRS</invr:granularity>
       </invr:includedObjectType>
       <invr:includedObjectType>
         <invr:objectType>CTP</invr:objectType>
         <invr:granularity>ATTRS</invr:granularity>
       </invr:includedObjectType>
     </invr:filter>
   </invr:getInventoryRequest>
 </soap:Body>
</soap:Envelope>
```

Figure 15. SFB MEP: getInventory request
(File “8-1_sfb_getInventoryRequest.xml” – MRI DDP)

```
 <?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:hdr="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1" xmlns:invr="http://www.tmforum.org/mtop/mri/xsd/rir/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tmforum.org/mtop/mri/xsd/rir/v1 ../xsd/ResourceInventoryRetrievalMessages.xsd
://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
   <hdr:header>
     <hdr:activityName>getInventory</hdr:activityName>
     <hdr:msgName>getInventoryResponse</hdr:msgName>
     <hdr:msgType>RESPONSE</hdr:msgType>
     <hdr:senderURI>/MTOSI/EMS01</hdr:senderURI>
     <hdr:destinationURI>/MTOSI/InventoryOS</hdr:destinationURI>
     <hdr:communicationPattern>BulkResponse</hdr:communicationPattern>
     <hdr:communicationStyle>RPC</hdr:communicationStyle>
     <hdr:timestamp>2007-07-30T09:26:00</hdr:timestamp>
   </hdr:header>
 </soap:Header>
 <soap:Body>
   <invr:getInventoryResponse/>
 </soap:Body>
</soap:Envelope>
```

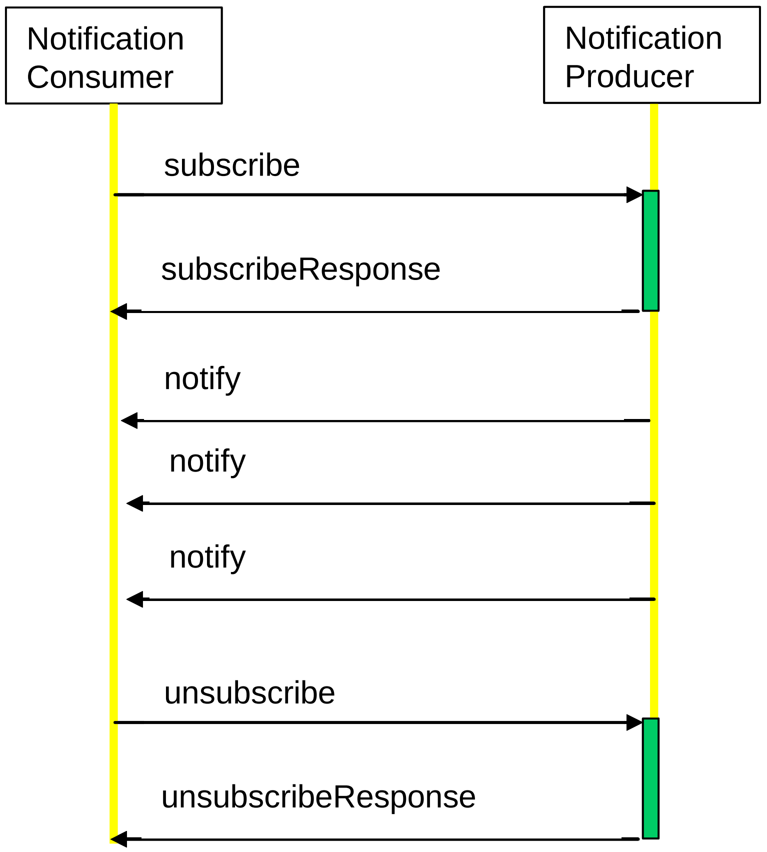
Figure 16. SFB MEP: getInventory response (acknowledgement)
(File “8-2_sfb_getInventoryResponse.xml” – MRI DDP)

3.5 Notification MTOSI Communication Pattern

As presented in [2], MTOSI supports two notification communication patterns: Brokered Notifications and Direct Notifications.

Brokered notifications relates more to the JMS binding while the Direct notifications is more appropriate for HTTP binding, but both patterns may be implemented with the two bindings.
MTOSI mandates that when HTTP is used as transport technology, at least the Direct Notifications pattern is implemented.

The dialogue sequence in the Direct Notifications is shown below (see [2]):



The notification subscriber must support a NotificationConsumer interface with its single operation “notify”.
The notification publisher must support a NotificationProducer interface with its “subscribe” and “unsubscribe” operations.

```
 <?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML request file of the MTOSI NotificationProducer/Broker subscribe operation for all FAULT notifications --
>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1 ../xsd/NotificationMessages.xsd
http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <m:header xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1">
     <m:activityName>Notification</m:activityName>
     <m:msgName>subscribeRequest</m:msgName>
     <m:msgType>REQUEST</m:msgType>
     <m:senderURI>http://www.altova.com</m:senderURI>
     <m:destinationURI>http://www.altova.com</m:destinationURI>
     <m:security>String</m:security>
     <m:securityType>String</m:securityType>
     <m:priority>String</m:priority>
     <m:msgSpecificProperties>
       <m:property>
         <m:propName>String</m:propName>
         <m:propValue>String</m:propValue>
       </m:property>
     </m:msgSpecificProperties>
     <m:communicationPattern>SimpleResponse</m:communicationPattern>
     <m:communicationStyle>RPC</m:communicationStyle>
     <m:timestamp>2001-12-17T09:30:47.0Z</m:timestamp>
     <m:vendorExtensions/>
   </m:header>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <m:subscribeRequest xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1">
     <m:consumerEpr>http://RequestingOs/NotificationConsumer</m:consumerEpr>
     <m:topic>FAULT</m:topic>
   </m:subscribeRequest>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 17. WSN MEP: subscribe request
(File “subscribeRequest-Fault.xml” – FMW DDP)

```
 <?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML response file of the MTOSI NotificationProducer/Broker subscribe operation for all FAULT notifications
-->
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1 ../xsd/NotificationMessages.xsd
http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
  <m:header xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1">
   <m:activityName>Notification</m:activityName>
   <m:msgName>subscribeResponse</m:msgName>
   <m:msgType>RESPONSE</m:msgType>
   <m:senderURI>http://www.altova.com</m:senderURI>
   <m:destinationURI>http://www.altova.com</m:destinationURI>
   <m:security>String</m:security>
   <m:securityType>String</m:securityType>
   <m:priority>String</m:priority>
   <m:msgSpecificProperties>
    <m:property>
     <m:propName>String</m:propName>
     <m:propValue>String</m:propValue>
    </m:property>
   </m:msgSpecificProperties>
   <m:communicationPattern>SimpleResponse</m:communicationPattern>
   <m:communicationStyle>RPC</m:communicationStyle>
   <m:timestamp>2001-12-17T09:30:47.0Z</m:timestamp>
   <m:vendorExtensions/>
  </m:header>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
  <m:subscribeResponse xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1">
   <m:subscriptionID>String</m:subscriptionID>
  </m:subscribeResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 18. WSN MEP: subscribe response
(File “subscribeResponse.xml” – FMW DDP)

```
 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1 ../xsd/NotificationMessages.xsd
http://www.tmforum.org/mtop/fmw/xsd/hbt/v1 ../xsd/EventHeartbeat.xsd http://schemas.xmlsoap.org/soap/envelope/
http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <m:header xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1">
     <m:activityName>Notification</m:activityName>
     <m:msgName>notify</m:msgName>
     <m:msgType>NOTIFICATION</m:msgType>
     <m:senderURI>http://www.altova.com</m:senderURI>
     <m:destinationURI>http://www.altova.com</m:destinationURI>
     <m:security>String</m:security>
     <m:securityType>String</m:securityType>
     <m:priority>String</m:priority>
     <m:msgSpecificProperties>
       <m:property>
         <m:propName>String</m:propName>
         <m:propValue>String</m:propValue>
       </m:property>
     </m:msgSpecificProperties>
     <m:communicationPattern>Notification</m:communicationPattern>
     <m:communicationStyle>RPC</m:communicationStyle>
     <m:timestamp>2001-12-17T09:30:47.0Z</m:timestamp>
     <m:vendorExtensions/>
   </m:header>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <m:notify xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1">
     <m:topic>String</m:topic>
     <m:message>
       <m0:heartbeat xmlns:m0="http://www.tmforum.org/mtop/fmw/xsd/hbt/v1"
xmlns:m1="http://www.tmforum.org/mtop/fmw/xsd/cei/v1" xmlns:m2="http://www.tmforum.org/mtop/fmw/xsd/ei/v1">
         <m1:notificationId>String</m1:notificationId>
         <m1:sourceTime>2001-12-17T09:30:47.0Z</m1:sourceTime>
         <m1:vendorExtensions/>
         <m2:objectType>OS</m2:objectType>
         <m2:objectName>
           <nam:rdn xmlns:nam="http://www.tmforum.org/mtop/fmw/xsd/nam/v1">
             <nam:type>OS</nam:type>
             <nam:value>string</nam:value>
           </nam:rdn>
         </m2:objectName>
         <m2:osTime>2001-12-17T09:30:47.0Z</m2:osTime>
       </m0:heartbeat>
     </m:message>
   </m:notify>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 19. WSN MEP: notify
(File "notify-Heartbeat.xml" – FMW DDP)

```
 <?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML request file of the MTOSI NotificationProducer/Broker subscribe operation for all FAULT notifications --
>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1 ../xsd/NotificationMessages.xsd
http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <m:header xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1">
     <m:activityName>Notification</m:activityName>
     <m:msgName>unsubscribeRequest</m:msgName>
     <m:msgType>REQUEST</m:msgType>
     <m:senderURI>http://www.altova.com</m:senderURI>
     <m:destinationURI>http://www.altova.com</m:destinationURI>
     <m:security>String</m:security>
     <m:securityType>String</m:securityType>
     <m:priority>String</m:priority>
     <m:msgSpecificProperties>
       <m:property>
         <m:propName>String</m:propName>
         <m:propValue>String</m:propValue>
       </m:property>
     </m:msgSpecificProperties>
     <m:communicationPattern>SimpleResponse</m:communicationPattern>
     <m:communicationStyle>RPC</m:communicationStyle>
     <m:timestamp>2001-12-17T09:30:47.0Z</m:timestamp>
     <m:vendorExtensions/>
   </m:header>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <m:unsubscribeRequest xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1">
     <m:subscriptionID>String</m:subscriptionID>
   </m:unsubscribeRequest>
 </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

Figure 20. WSN MEP: unsubscribe request
(File “unsubscribeRequest.xml” – FMW DDP)

```
 <?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML request file of the MTOSI NotificationProducer/Broker subscribe operation for all FAULT notifications --
>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xsi:schemaLocation="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1 ../xsd/NotificationMessages.xsd
http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <m:header xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/hdr/v1">
     <m:activityName>Notification</m:activityName>
     <m:msgName>unsubscribeResponse</m:msgName>
     <m:msgType>RESPONSE</m:msgType>
     <m:senderURI>http://www.altova.com</m:senderURI>
     <m:destinationURI>http://www.altova.com</m:destinationURI>
     <m:security>String</m:security>
     <m:securityType>String</m:securityType>
     <m:priority>String</m:priority>
     <m:msgSpecificProperties>
       <m:property>
         <m:propName>String</m:propName>
         <m:propValue>String</m:propValue>
       </m:property>
     </m:msgSpecificProperties>
     <m:communicationPattern>SimpleResponse</m:communicationPattern>
     <m:communicationStyle>RPC</m:communicationStyle>
     <m:timestamp>2001-12-17T09:30:47.0Z</m:timestamp>
     <m:vendorExtensions/>
   </m:header>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <m:unsubscribeResponse xmlns:m="http://www.tmforum.org/mtop/fmw/xsd/notmsg/v1"/>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figure 21. WSN MEP: unsubscribe response
(File “unsubscribeResponse.xml” – FMW DDP)

4 References

[1] SD2-5, MTOSI Communication Styles

[2] SD2-9, MTOSI Using JMS as MTOSI Transport

[3] SOAP Version 1.1
W3C Note 08 May 2000
http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[4] Basic Profile Version 1.1
Final Material, 2006-04-10
http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[5] W3Schools, SOAP Tutorial
http://www.w3schools.com/soap/

[6] HTTP 1.1 Header Field Definitions
http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

5 Appendix: The HTTP V1.1 Header

The following table summarizes the HTTP Header fields extracted from the RFC 2616 [6].

General Header fields are of general applicability for both request and response message, but do not apply to the entity being transferred. They apply only to the message being transmitted.

Entity Header fields define meta-information about the entity-body or, if no body is present, about the resource identified by the request.

Field Name	Category	Description
Accept	Request	Specifies which Internet media types are acceptable for the response and to assign preferences to them
Accept-Charset	Request	Specifies which character encodings (confusingly called "charsets") are acceptable for the response and to assign preferences to them.
Accept-Encoding	Request	Specifies which data format tranformations, confusingly called content (en)codings, such as compression mechanisms, are acceptable for the response and to assign preferences to them.
Accept-Language	Request	Specifies which natural languages are acceptable for the response and to assign preferences to them. Useful for language negotation.
Accept-Ranges	Response	Indicates the server's acceptance of range requests for a resource.
Age	Response	Gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.
Allow	Entity	Lists the set of methods supported by the resource identified by the Request-URI. The purpose is to inform the recipient of valid methods associated with the resource.
Authorization	Request	Consists of credentials containing the authentication information of the client for the realm of the resource being requested
Cache-Control	General	Specifies directives that *must* be obeyed by all caching mechanisms along the request/response chain.
Connection	General	Specifies options that are desired for the particular connection and *must not* be communicated by proxies over further connections.
Content-Encoding	Entity	Used as a modifier to the media-type, to indicate what additional data format transformations such

		as compression have been applied to the entity-body.
Content-Language	Entity	Specifies the natural language(s) of the intended audience for the enclosed entity. But according to RFC 3282, specifies the language(s) of the entity.
Content-Length	Entity	Indicates the size (in octets) of the entity-body that is sent or that would have been sent if it has reen requested.
Content-Location	Entity	Supplies the resource location for the entity enclosed in the message when that entity is accessible from a location separate from the requested resource's URI.
Content-MD5	Entity	An MD5 digest of the entity-body for the purpose of providing an end-to-end message integrity check (MIC) of the entity-body.
Content-Range	Entity	Sent with a partial entity-body to specify where in the full entity-body the partial body should be applied.
Content-Type	Entity	Specifies the Internet media type of the entity-body that is sent or would have been sent if requested. Often includes a `charset` parameter specifying the character encoding.
Date	General	Date and time at which the message was originated.
ETag	Response	Provides the current value of the entity tag for the requested variant, for caching purposes.
Expect	Request	Indicates that particular server behaviors are required by the client.
Expires	Entity	Gives the date/time after which the response is considered stale, for caching purposes.
From	Request	The Internet e-mail address for the human user who controls the requesting browser or other client.
Host	Request	Specifies the Internet host and port number of the resource being requested. Obligatory in all HTTP/1.1 requests.
If-Match	Request	Used with a method to make it conditional: a client that has previously obtained entities can verify that one of those entities is current by including a list of their associated entity tags in the `If-Match` header field.
If-Modified-Since	Request	Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.

If-None-Match	Request	Used with a method to make it conditional: a client that has previously obtained entities can verify that none of those entities is current by including a list of their associated entity tags in the `If-None-Match` header field.
If-Range	Request	Used together with `Range` to say: "if the entity is unchanged, send me the part(s) that I am missing; otherwise, send me the entire new entity".
If-Unmodified-Since	Request	Used with a method to make it conditional: if the requested variant has been modified since the time specified in this field, the server will not perform the requested operation but information about this fact.
Last-Modified	Entity	Indicates the date and time at which the origin server believes the variant was last modified.
Location	Response	Redirects the recipient to a location other than the Request-URI for completion of the request or identification of a new resource
Max-Forwards	Request	Provides a mechanism with the `TRACE` and `OPTIONS` methods to limit the number of proxies or gateways that can forward the request to the next inbound server.
Pragma	General	Used to include implementation-specific directives that might (optionally) apply to any recipient along the request/response chain.
Proxy-Authenticate	Response	Included as part of a 407 (Proxy Authentication Required) response. The field value consists of a challenge that indicates the authentication scheme and parameters applicable to the proxy for this Request-URI.
Proxy-Authorization	Request	Used by a client to identify itself (or its user) to a proxy which requires authentication.
Range	Request	Restricts the request to some part(s), specified as range(s) of octets, in the resource.
Referer	Request	Used by a client to specify, for the server's benefit, the address (URI) of the resource from which the Request-URI was obtained.
Retry-After	Response	Indicates how long the service is expected to be unavailable to the requesting client.
Server	Response	Contains information about the software used by the origin server to handle the request.
TE	Request	Indicates what extension transfer-codings the client is willing to accept in the response and whether or not it is willing to accept trailer fields in a chunked transfer-coding.

Trailer	General	Indicates that the given set of header fields is present in the trailer of a message encoded with chunked transfer-coding.
Transfer-Encoding	General	Indicates what (if any) type of transformation has been applied to the message body in order to safely transfer it between the sender and the recipient. This differs from the `Content-Encoding` in that the transfer-coding is a property of the message, not of the entity.
Upgrade	General	Used by a client to specify what additional communication protocols it supports and would like to use if the server finds it appropriate to switch protocols. The server uses the `Upgrade` header to indicate which protocol(s) are being switched.
User-Agent	Request	Contains information about the user agent (client) originating the request
Vary	Response	Indicates the set of request-header fields that fully determines, while the response is fresh, whether a cache is permitted to use the response to reply to a subsequent request without revalidation.
Via	General	Used by gateways and proxies to indicate the intermediate protocols and recipients between the user agent and the server on requests, and between the origin server and the client on responses.
Warning	General	Carries additional information about the status or transformation of a message which might not be reflected in the message.
WWW-Authenticate	Response	Used in 401 (Unauthorized) response messages. The field value consists of at least one challenge that indicates the authentication scheme(s) and parameters applicable to the Request-URI.

6 Administrative Appendix

6.1 Document History

Version	Date	Description of Change
1.0	Oct 2006	This is the first version of this document and as such, there are no revisions to report.
1.1	May 2008	Aligned with the MTOSI 2.0 artifacts

6.2 Acknowledgments

First Name	Last Name	Company
Francesco	Caruso	Telcordia
Basma	Driss	Ericsson
Jerome	Magnet	Nortel
Ramanathan	Krishnamurthy	IONA
Nathan	Sowatskey	Cisco

6.3 How to comment on this document

Comments and requests for information must be in written form and addressed to the contact identified below:

Michel	Besson	**Amdocs OSS**
Phone:	+44 7717 692 178	
Fax:		
e-mail:	Michel.Besson@Amdocs.com	

Please be specific, since your comments will be dealt with by the team evaluating numerous inputs and trying to produce a single text. Thus we appreciate significant specific input. We are looking for more input than wordsmith" items, however editing and structural help are greatly appreciated where better clarity is the result.