STAR OAGI 9 BOD

Implementation Guidelines

Version 1.1

March 2008

Revision History

Revision Date	Version	Description
Mar. 30, 2007	1.0	Initial Version
Mar. 03, 2008	1.1	* Noted the deprecation of the versionID attribute in version 5.0.2 * Added section on Correlating Nouns with Release Numbers

Table of Contents

STAR BOD Implementation Reference

1. INTRODUCTION

The Standards for Technology in Automotive Retail (STAR) organization is a non-profit organization that develops open, voluntary standards for the retail automotive industry. The extensible mark-up language (XML) standards that STAR creates are referred to as Business Object Documents (BODs) and are based on the Open Applications Group (OAG) development methodology. OAG is a not-for-profit standards consortium whose purpose is to promote interoperability among business software applications and to create and/or endorse standards for improved business software interoperability.

STAR has developed over 125 BODs. The STAR BODs are developed to support multiple areas of business including customer relationship management, parts management, vehicle management, service and repairs, warranty, etc.

The purpose of this document is to define general information to assist developers in understanding STAR BODs, how BODs are designed, their intended use, and what is considered to be compliant. This document may be used to assist in the implementation of STAR BODs.

NOTE: This document applies to the STAR 5.0 Schema Repository and forward. It does not apply to any repository prior to STAR 5.0

2. UNDERSTANDING KEY CONCEPTS

Prior to implementing a STAR BOD, implementers should become familiar with several concepts.

2.1. UNDERSTANDING XML

STAR BODs are based on XML. Therefore it is important to have a good understanding of XML and schema. There are several online resources freely available as well as reference books. One suggested online resource for understanding XML is www.XML.com. If you are implementing Web Services, another suggested resource is http://webservices.xml.com/.

2.2. UNDERSTANDING YOUR TOOLS

It is important to understand the tool(s) that you will be using to implement and that tool(s) is suited for working with XML. At a minimum, a tool must support the W3C Schema Specification version 1.0. If the tool does not support this, it will make implementation of the STAR BODs challenging. If you are using a version of a tool that does not support the W3C Schema Specification version 1.0, you may need to consider updating the tool. A copy of the W3C Schema Specification can be found by going to www.w3.org/XML/Schema.

Please note that the standards that STAR creates are tool agnostic and are not required to support a specific tool or version of a tool. STAR will however be providing a STAR Workbench in 2007 that will be an open source, entry level tool for importing and viewing the STAR schema.

2.3. UNDERSTANDING STAR

Implementers should also familiarize themselves with the STAR organization and the STAR BOD methodology. It is important to understand how the BODs are developed, the publication schedules, the modification request process, the process for developing new BODs etc.

2.3.1. BOD Development Process

All STAR BODs are developed using the Open Application Group's BOD methodology. Where possible, STAR leverages:

- Existing OAGI components, fields and BODs

- Existing core components defined by the UN/CEFACT

- Existing components and fields defined by the Joint Automotive Data Model – a global initiative between several automotive industry organizations including STAR

BODs are developed based on value statements submitted by member organizations. **STAR only develops BODs for members** and those members must have a target implementation of within 6 months to a year. Members are however encouraged to use existing STAR BODs where possible. In some cases a particular BOD may require modifications to meet a member's business requirements.

2.3.2. BOD Maintenance

STAR promotes a flexible change control policy for updating BODs. Members may submit modification requests for a BOD at any time with the following requirements:

- The submitter must be a STAR Member

- A modification request form is completed. For a copy of STAR's BOD modification request form go to: http://www.starstandard.org/index.php?n=STARLINKS.Forms

- A planned implementation within 6 months to a year

STAR does not support the use of "user areas" or extensions. If a member finds that their business requirements are not being met by a particular BOD, they are encouraged to submit a modification request rather than updating the BOD themselves.

2.3.3. BOD Publication Schedule

STAR BODs are officially published and made publicly available once a year in May with an effective date of July 4th. The BODs are published to www.starstandard.org .

For members, draft BODs are published throughout the year in the following forms:

- Milestones: These are snapshots of the repository at a given time and are published every two weeks

- Release Candidates: These are published towards the end of a draft release cycle to perform any bug fixes.

- Draft Releases – These are draft versions of the BODs that are published 4 times per year. Members are permitted to implement draft versions but are cautioned that they may change between the draft and the final release in July.

2.4. UNDERSTANDING BODS

BODs are comprised of 6 segments of information:

BOD

Figure 1 Segments of a BOD

2.4.1. Application Area

The Application Area provides the information that an application may need to know in order to communicate in an integration of two or more business applications. The Application Area is used at the application layer of communication. While the integration framework's web services and middleware provide the communication layer on top of which STAR operates.

The Application Area is a common component that is used in all STAR BODs. Therefore, the information is generic in nature and does not relate specifically to any one BOD. For this reason, very rarely are fields added to the Application Area.

There are five sections of information contained in the Application Area:

1. Sender

2. CreationDateTime

3. Signature

4. BODId

5. Destination

2.4.2. Data Area

The Data Area of the BOD is comprised of the Verb and the Noun or Nouns.

2.4.3. Verbs

The Verb is the portion of the BOD that identifies the action to be taken. For example, in the BOD ProcessPartsOrder, the Verb being applied is Process which indicates that once the message is received it is intended to be processed. A BOD can only contain one definition of a Verb. To indicate a different action a separate BOD would need to be defined. For example, in order for a receiver of a ProcessPartsOrder message to respond that they have received and processed the message they would send back an **Acknowledge**PartsOrder.

All of the Verbs that STAR uses are defined by OAGI and are imported from the OAGIS schema repository. OAGI has defined the usage of all Verbs and how they are to be paired in collaboration. STAR does not support pairings of Verbs that do not follow the OAGIS Verb usage guidelines

2.4.4. Nouns

The Noun is the portion of the BOD that identifies the business specific data using Components and individual fields that is being communicated (i.e., PartOrder, SalesLead, VehicleServiceHistory, etc.). A BOD must have a minimum of one Noun, but may contain multiple Nouns. If multiple Nouns are transmitted in a single BOD, they must all be the same type of Noun.

The Noun is the object that is being acted upon by the Verb. Note that not all Verbs apply to all Nouns.   During development, STAR identifies a series of Verb/Noun combinations or BODs to satisfy a particular collaboration scenario(s).   These Verb/Noun combinations follow the Verb usage guidelines set by OAGI. If a particular scenario has not been covered by the existing set of BODs, a modification request may be submitted by a Member.

NOTE: STAR does not allow for extensions or modifications of any kind to Nouns or Verbs. STAR also does not all for the pairing of Nouns and Verbs that have not been approved by STAR.

2.4.5.   STAR Release Numbers

Each publication of the STAR XML Repository will have a unique Release number using the format "nn.n" beginning with "1.0".   Interim versions of a Release are identified by incrementing the published version of the modified Release by ".01". (i.e.; "1.0.1").

If a Release is considered to be a Major Release, i.e., there are changes that will impact forward compatibility, the Release will be versioned by 1.0. If a Release is considered to be a Minor Release, i.e. there are no changes that would impact forward compatibility; the Release will be versioned by .1.

The Release Number of a BOD can be found in the metadata of the XML file that is generated by a BOD. The attribute is called **releaseID** and is required.

```
<AcknowledgeCreditApplication releaseID="5.0.4" versionID="6.0" systemEnvironmentCode="Production"
languageCode="en-US" xmlns:ns="http://www.starstandard.org/STAR/5"
xmlns:ns1="http://www.openapplications.org/oagis/9">
```

NOTE: The **versionID** attribute has been deprecated as of version 5.0.2.

2.4.5.1.   STAR Namespace

The Release Number of the STAR Schema Repository is also captured in the STAR namespace.

xmlns:ns=http://www.starstandard.org/STAR/**5**

If a Release is considered to be a Major Release, the release number in the STAR namespace will be incremented by 1.0. Note that any change to a namespace impacts forward compatibility. Therefore, changes to a namespace would also indicate a Major Release.

If a Release is considered to be a Minor Release, the release number in the STAR namespace will remain the same.

2.4.5.2. Correlating Nouns with Release Numbers

STAR does not version individual Nouns. Instead, Nouns are correlated with the releaseID of the repository in which they appear. For example, all Nouns in STAR Schema Repository 5.0.4 will are referenced using releaseID, "5.0.4". So the Parts Order noun appearing in 5.0.4 would be referenced as PartsOrder 5.0.4.

2.4.5.3. OAGI Release Number

The OAGI release that a particular BOD is based on is identified within the OAGIS namespace found in the XML instance of a BOD.

```
xmlns:oagis=http://www.openapplications.org/oagis/9
```

If a Release is considered to be a Major Release, the release number in the OAGIS namespace will be incremented by 1.0. Note that any change to a namespace impacts forward compatibility. Therefore, changes to a namespace would also indicate a Major Release.

If a Release is considered to be a Minor Release, the release number in the OAGIS namespace will remain the same.

3. STAR BOD DELIVERABLES

There are two major deliverables for each STAR BOD:

- STAR Implementation Guidelines
- STAR Schema Repository

3.1. STAR IMPLEMENTATION GUIDELINES

This document is a Guideline on how to implement a BOD using the STAR schema. It explains the relationships, definitions, and valid values of the data. Users can review the guidelines to understand how their requirements are being addressed. Developers can use the guidelines to determine how to map STAR requirements to their own internal requirements. The guidelines are available in Portable Document Format (PDF), Hypertext Markup Language (HTML) and OpenDocument Text (ODT) formats.

3.1.1. Columns in the Component Tables

This is where the detailed definition on the BOD data requirements is contained. There are several columns in each component table:

- **Field Name/Component** – This is the actual tag that corresponds to the schema. If a component rather than a field or compound is defined in this column, a notation will be in the "Comment" column saying "See xxxx Component"

- **Definition** - The definition of the field. If for a component, this column is usually left blank.

- **Required/ Optional** – STAR tries to define most fields as optional to keep the standard open and flexible. If a field is "Required, it must be populated when the XML is created. If a field is "Optional", but it is required for a specific organization, it will be their responsibility to make it "Required" within their application.

- **Valid Values or "Qualifiers"** – This is where any enumerators associated with a field are defined. It would be a qualifier if a component was being defined in the first column "Field Name/Component"

- **Business Rules/ Comments** – Comments relating to field, compound or component. When attributes are used on a compound, they will be listed here. This column can also contain examples of how data would be formatted and Business Rules that apply to the implementation of the field, compound or component.

Field/Component	Description	Requirement	Business Rules
DocumentDateTime	Is the date and time the document was last created. This is not the date and time that the BOD message instance was created.	Optional	
SecondaryPassword	Secondary password used to validate access to the dealer information	Optional	
SecondaryDealerNumberID	Identifies secondary dealer number if different than primary "Dealer Number"	Optional	

Figure 2 BOD Implementation Guideline Table

3.2. STAR SCHEMA REPOSITORY

The STAR Schema Repository contains the Schema and Sample XML for all the approved BODs.

3.2.1. Sample XML

Sample XML will provide the developers assistance in understanding how XML will look. It contains examples of the actual data tags and some sample data. It is populated with one example when valid values are specified. Note however the sample XML that is provided in the STAR schema repository is not required to validate. It is up to the end user to generate their own sample XML file using their tool set.

3.2.2. Schema Repository Location and Access

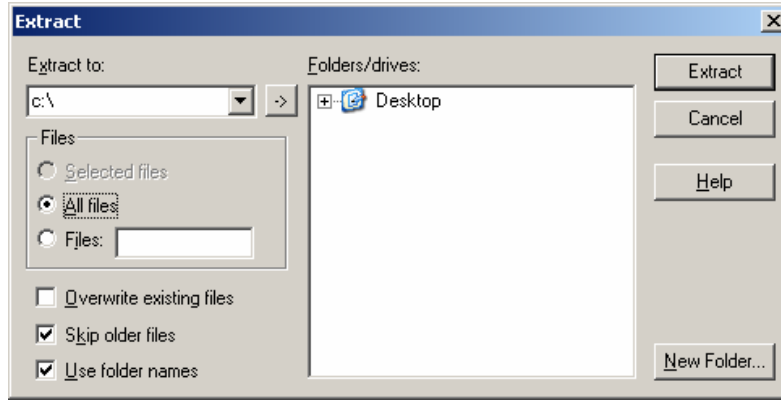Schema is located on the STAR web site: www.STARStandard.org.

Figure 3 How to Extract the STAR Schema Repository

Download the schema zip file "STAR" and extract it making sure to take the option to "Use Folder Names" from WinZip or any other Zip compatible program.

The STAR Repository and structure is setup to be Tool independent. Only the directory structure must remain intact. If you are using one of the available XML Integrated Development Environments (IDEs) please refer to that IDE's documentation on how to import the schemas into your particular environment.

Figure 4 STAR Schema Repository Structure

4. DATA COMPLIANCE SUMMARY

The following section defines STAR compliance at the data level.

4.1. WHITESPACE

Ignore the Whitespace that is only there for readability. It should not be in what goes over the wire.

4.2. FIELD POPULATION

The following sections define how fields are to be populated.

4.2.1. Optional Fields

Optional fields do not need to be sent over the wire. Empty tags should not be sent. Sender must send all optional fields available in their system, and the Receiver will be responsible for only recognizing fields that they need. An out-of-band negotiation is required if a receiver only wants to:

- Receive a subset of the optional fields
- Request data not available in the sender's system
- Request validation of non-enumerated fields

Note: an out-of-band negotiation would not be considered "non-compliant".

4.2.2. Required Fields

A required field must be populated with a value. Null values are not allowed. A required field on a component is only required to be sent when the component is sent. If an enumerated value has been defined for a required field, then the sender would need to populate the field with one of the defined enumerators.

4.2.3. All Fields

All fields that are sent are required to contain data, whether the field is required or optional. A field will not validate if it contains:

- Null values
- Spaces

4.3. EXTENSIONS

The following sections define how extensions are managed in STAR BODs.

4.3.1. Noun Extensions

Extensions to STAR Nouns via Users Areas or additional fields are not allowed. If additional requirements are needed, a member may submit a modification request at any time to have them added.

4.3.2. CodeList Extensions

Extensions to STAR CodeLists, i.e. enumerated lists, are not allowed. If additional requirements are needed, a member may submit a modification request at any time to have them added.

4.4. VALIDATION

XML instances must validate against the appropriate STAR BOD schema. XML instances must also be validated using the "strict" schema validation option of the parser. The "lax" option is not star compliant.

5. VERB USAGE & PROCESS FLOWS

The following verb process flow diagrams are the standard default process flows defined by OAGIS for implementing STAR BODs. STAR recognizes that each implementation of a STAR BOD will be unique depending on the chosen transport, chosen software, etc. As a result, the associated sequence diagrams have been developed at a high level to allow room for individual implementations.

Note: If a member chooses to implement a process flow other than those identified by STAR, that implementation would be considered custom. If a member's process flow is not covered by the STAR standard process flow, a modification request should be submitted to STAR for review and possible inclusion.

5.1. PROCESS & ACKNOWLEDGE

The following verb usage description has been defined by OAGIS:

"Use Process to request creation of a new entity or "process" a change to an object that may or may not already be an existing entity. The sender doesn't know whether the entity already exists or not. The Process is used to communicate a request for something to be done between peer systems or from a system that does not own the given Noun to the owner of the data or to another peer.

This Process may result in an Acknowledge response. When the Process is used to communicate between peer system the Acknowledge is required.

It should be noted that the Process and Acknowledge pair are used to communicate requests and acknowledge responses between different legal entities to form a binding agreement as in the case of a PurchaseOrder. The ProcessPurchaseOrder asks a supplier to fulfill the order the positive AcknowledgePurchaseOrder accepts the order and legal binds the supplier to fulfill the PurchaseOrder." (Open Applications Group 1)
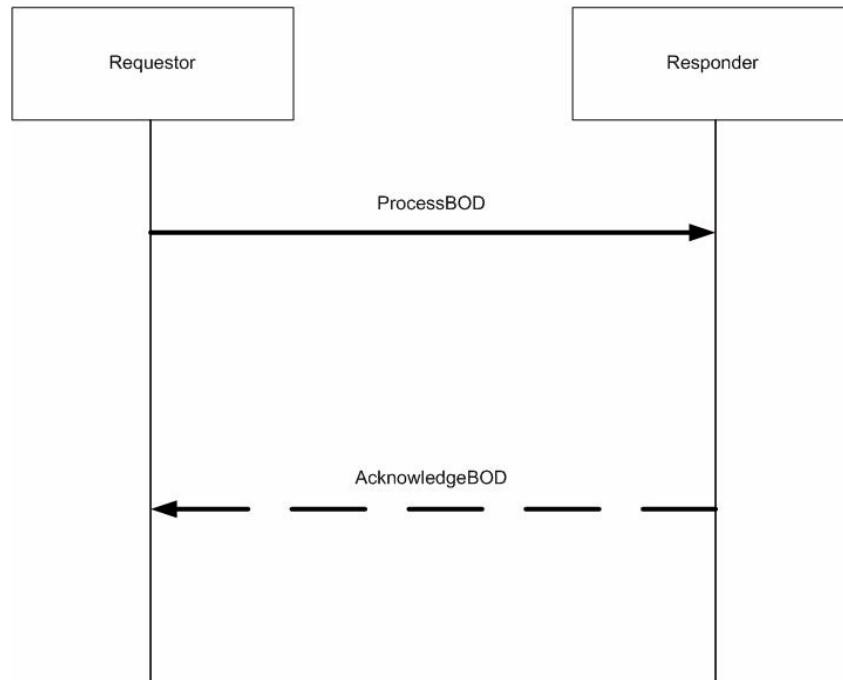
Process & Acknowledge

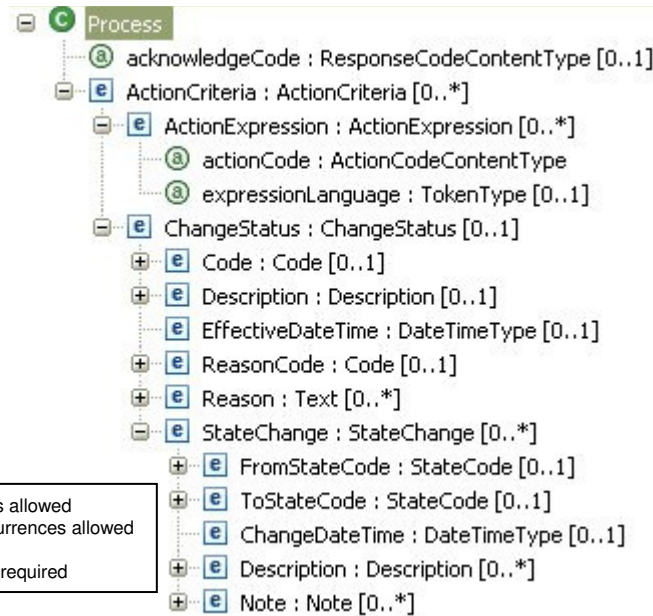**Figure 5 Process & Acknowledge Sequence Diagram**

5.1.1. *Process*

OAGIS has defined the Process verb as follows:

"The Process verb is used to request processing of the associated noun by the receiving application or business to party. In a typical external exchange scenario a Process BOD is considered to be a legally binding message. For example, if a customer sends a ProcessPartsOrder BOD to a supplier and the supplier acknowledges with a positive AcknowledgePartsOrder, then the

customer is obligated to fulfill the agreement, unless of course other BODs are allowed to cancel or change the original order." (OAGIS 9.0 Meta.xsd)

Figure 6 Process Verb Attributes

ProcessVerb	Definition
acknowledgeCode	Enumerated attributed used to indicate when an Acknowledge should be returned. Values: "Always" "Never" "OnError"
ActionCriteria	Identifies with the ActionExpression the level by carrying a value of the expressionLanguage (this is typically XPath) within the BOD that the actionCode is to be performed. The ChangeStatus communicates just that

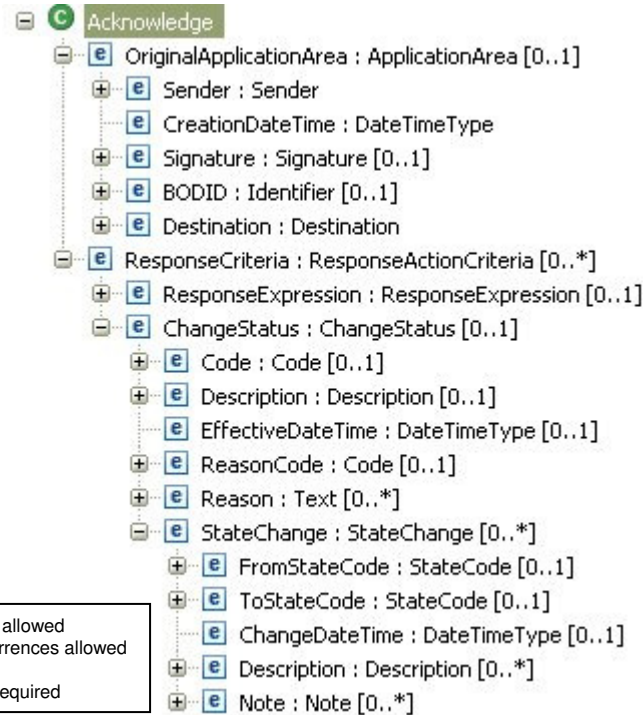|	the ChangeStatus.
ActionExpression	ActionExpression identifies the content that is to be acted upon, given query success. In essence, the expression here has the effect of filtering the part(s) of the found element(s) that are to be acted upon. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 1 Process Verb Attributes Defined

5.1.2. *Acknowledge*

OAGIS defines the Acknowledge verb as follows:

"The Acknowledge verb is used to acknowledge the application receipt of a Process request. This function conveys the result of the original request. An example of this is AcknowledgePartsOrder, where a ProcessPartsOrder has been issued and the corresponding business application acknowledges the receipt of the PartsOrder and responds with an acceptance or a counter offer." (OAGIS 9.0 Meta.xsd)

Figure 7 Acknowledge Verb Attributes

RespondVerb	Definition
OriginalApplicationArea	A copy of the ApplicationArea for the original BOD that was processed. Present either as additional reference information, or for use in identifying the BOD in situations where a BODReference is not known.
ResponseCriteria	ResponseCriteria identifies the content that is to be returned, given query success. In essence, the expression here has the effect of filtering the part(s) of the found element(s) that are to be returned. ReturnCriteria plays no role in the query itself. That is handled as a match against the request BOD's noun exemplar. ReturnCriteria allows the sender of the BOD to indicate which information (down to the field level) is requested to be returned, given that the query has been successful in matching the exemplar to existing nouns. That is, in a GetListPurchaseOrder, if one or more PurchaseOrders with a TotalPrice = $1M were found, ReturnCriteria tells the BOD recipient which parts of the PurchaseOrder should be

	populated with content when the response (ShowPurchaseOrder) is formulated. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 2 Acknowledge Verb Attributes Defined

5.2. CANCEL & CONFIRM BOD

The following verb usage description has been defined by OAGIS:

"Use Cancel to cancel an earlier order/request. The Cancel can be sent by either side as long as a legally binding Acknowledge have not been sent. This results in a ConfirmBOD response." (Open Applications Group 1)
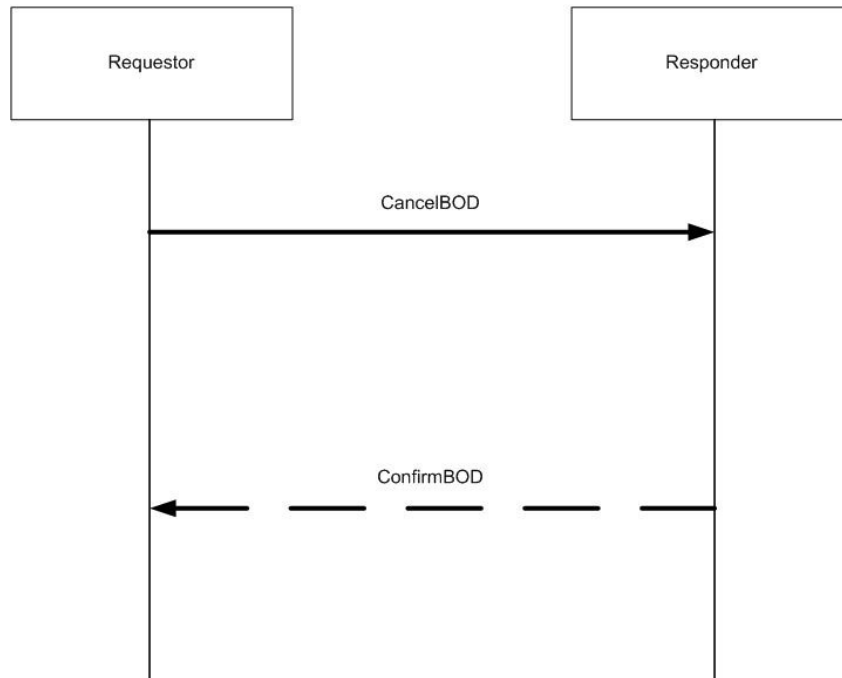
Cancel & Confirm

Figure 8 Cancel & Confirm Sequence Diagram

5.2.1. *Cancel*

OAGIS defines the Cancel verb as follows:

"The Cancel verb is used when the sender of the BOD is not the owner of the data, but is sending a request for the document to be canceled. An example is the Cancel PO where the business implications must be calculated and a simple data processing term such as delete can not fully convey the business meaning and required processing associated with the meaning." (OAGIS 9.0 Meta.xsd)
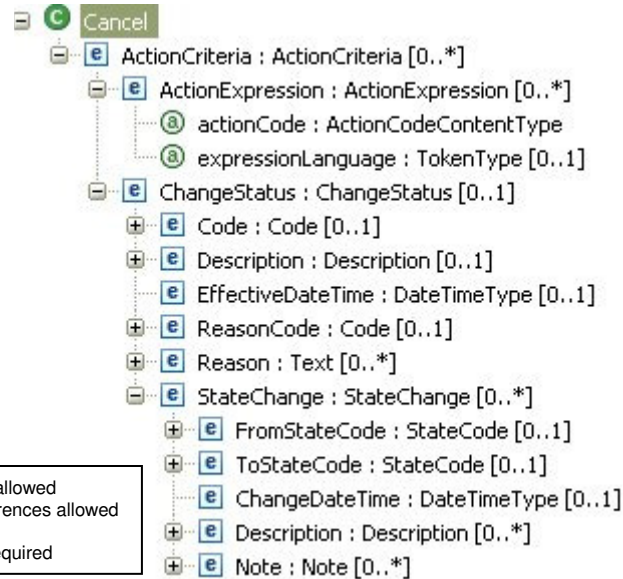
STAR

[0..1] = 1 optional occurrences allowed
[0..*] = 0 to many optional occurrences allowed
[1..1] = 1 occurrence required
[1..*] = 1 or more occurrences required

Figure 9 Cancel Verb Attributes

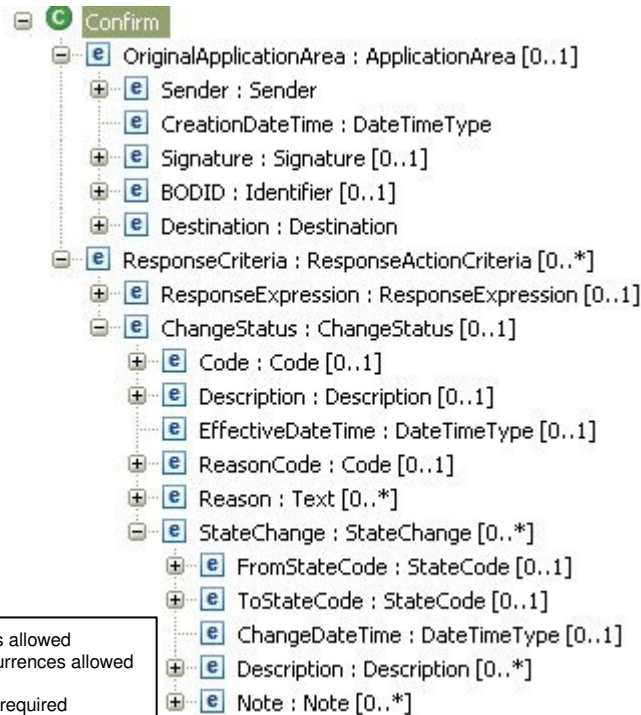| CancelVerb	Definition
ActionCriteria	Identifies with the ActionExpression the level by carrying a value of the expressionLanguage (this is typically XPath) within the BOD that the actionCode is to be performed. The ChangeStatus communicates just that the ChangeStatus.
ActionExpression	ActionExpression identifies the content that is to be acted upon, given query success. In essence, the expression here has the effect of filtering the part(s) of the found element(s) that are to be acted upon. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the

	change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 3 Cancel Verb Attributes Defined

5.2.2. Confirm

OAGIS defines the Confirm verb as follows:

"The Confirm verb is used to respond to a request to confirm the receipt of information by the receiving system. The request for confirmation is set by the sending application in the ApplicationArea\Sender\Confirmation field of the original BOD. The Confirm conveys the result of the original request i.e. whether or not the message was understood and was successfully processed. An example of this is Confirm BOD." (OAGIS 9.0 Meta.xsd)

Figure 10 Confirm Verb Attributes

ConfirmVerb	Definition
OriginalApplicationArea	A copy of the ApplicationArea for the original BOD that was processed. Present either as additional reference information, or for use in identifying the BOD in situations where a BODReference is not known.
ResponseCriteria	ResponseCriteria identifies the content that is to be returned, given query success. In essence, the expression here has the effect of filtering the part(s) of the found element(s) that are to be returned. ReturnCriteria plays no role in the query itself. That is handled as a match against the request BOD's noun exemplar. ReturnCriteria allows the sender of the BOD to indicate which information (down to the field level) is requested to be returned, given that the query has been successful in matching the exemplar to existing nouns. That is, in a GetListPurchaseOrder, if one or more PurchaseOrders with a TotalPrice = $1M were found, ReturnCriteria tells the BOD recipient which parts of the PurchaseOrder should be

	populated with content when the response (ShowPurchaseOrder) is formulated. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 4 Confirm Verb Attributes Defined

5.3. CHANGE & RESPOND

The following verb usage description has been defined by OAGIS:

"Use Change Verb to request a change to an entity between peer system or if the request is going to the owner of the information and the sending system knows the information exists. If the change comes from the owner of the information Sync should be used. If the request is to the owner of the information the Process should be used.

This may results in an optional Respond Verb with the original Noun as the response where the Respond is used to further communicate the status of the Change.

The Change BOD communicates the full detail of the given Noun and should replace any existing information for the given entity. If the request only communicates the delta changes the Update Verb should be used." (Open Applications Group 1)
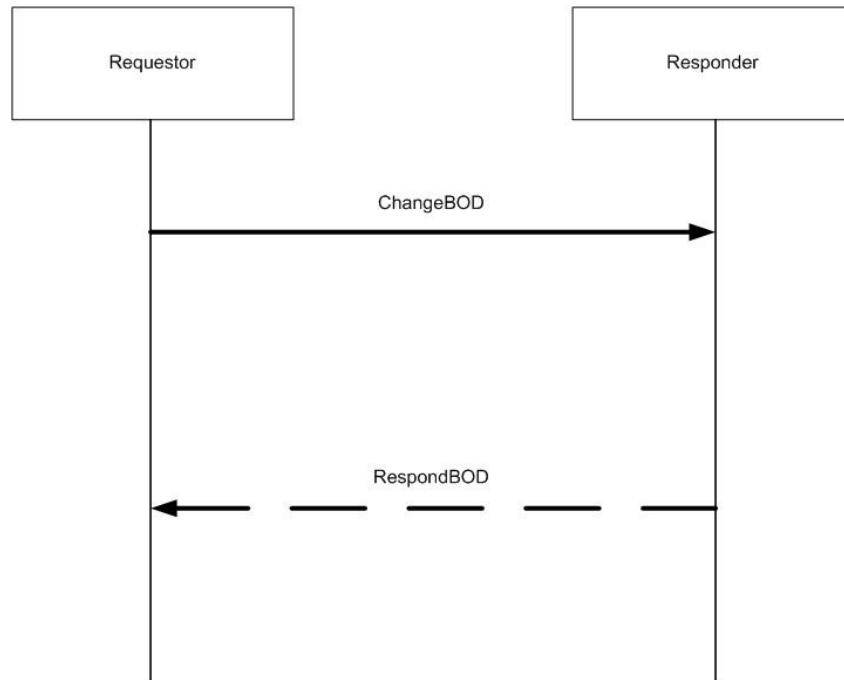
Change & Respond

Figure 11 Change & Respond Sequence Diagram

5.3.1. Change

OAGIS defines the Change verb as follows:

"The Change verb is used when the sender of the BOD is not the owner of the data, but is sending a request for the document to be changed. An example of this is Change PartsOrder, where the original document needs to be changed based on a specific business event." (OAGIS 9.0 Meta.xsd)
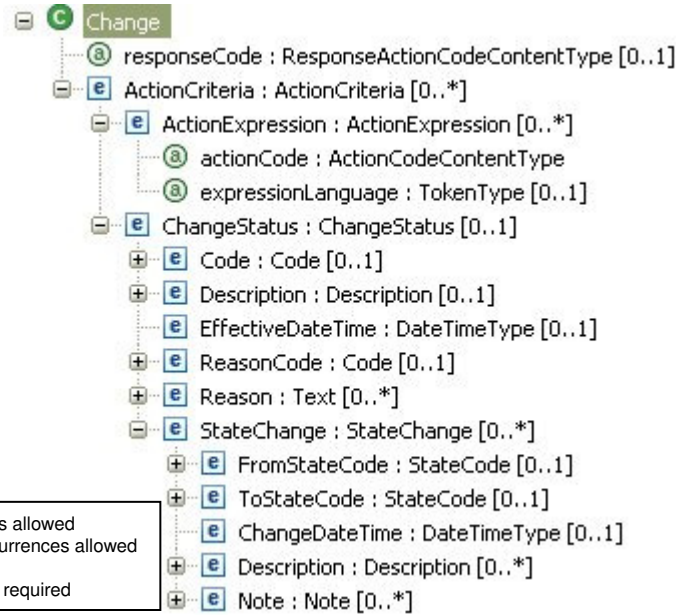
[0..1] = 1 optional occurrences allowed
[0..*] = 0 to many optional occurrences allowed
[1..1] = 1 occurrence required
[1..*] = 1 or more occurrences required

Figure 12 Change Verb Attributes

ChangeVerb	Definition
responseCode	Enumerated attributed used to indicate when a Respond BOD should be returned. Values: "Always" "Never" "OnError"
ActionCriteria	Identifies with the ActionExpression the level by carrying a value of the expressionLanguage (this is typically XPath) within the BOD that the actionCode is to be performed. The ChangeStatus communicates just that the ChangeStatus.
ActionExpression	ActionExpression identifies the content that is to be acted upon, given query success. In essence, the expression here has the effect of filtering

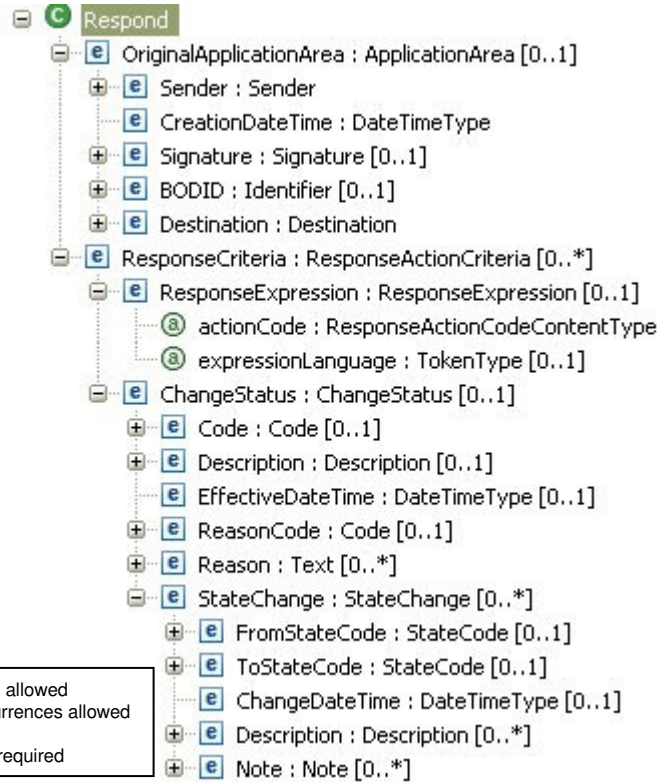|	the part(s) of the found element(s) that are to be acted upon. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 5 Change Verb Attributes Defined

5.3.2. Respond

OAGIS defines the Respond verb as follows:

"The Respond verb is used to communicate relative to another document. It may be used to communicate agreement, questions, answers to a question, or disagreement with the related document. An example is the change of a Parts Order. A Parts Order is issued to a set of business partners. If one of the partners needs to modify an item on the Parts Order, a RespondPartsOrder is sent to the originating partner communicating agreement with the change." (OAGIS 9.0 Meta.xsd)

Figure 13 Respond Verb Attributes

RespondVerb	Definition
OriginalApplicationArea	A copy of the ApplicationArea for the original BOD that was processed. Present either as additional reference information, or for use in identifying the BOD in situations where a BODReference is not known.
ResponseCriteria	ResponseCriteria identifies the content that is to be returned, given query success. In essence, the expression here has the effect of filtering the part(s) of the found element(s) that are to be returned. ReturnCriteria plays no role in the query itself. That is handled as a match against the request BOD's noun exemplar. ReturnCriteria allows the sender of the BOD to indicate which information (down to the field level) is requested to be returned, given that the query has been successful in matching the

	exemplar to existing nouns. That is, in a GetListPurchaseOrder, if one or more PurchaseOrders with a TotalPrice = $1M were found, ReturnCriteria tells the BOD recipient which parts of the PurchaseOrder should be populated with content when the response (ShowPurchaseOrder) is formulated. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 6 Respond Verb Attributes Defined

5.4. GET & SHOW

The following verb usage description has been defined by OAGIS:

"Use Get to get entity data. The system being requested to provide more information on the Noun receives and processes this message. This may or may not be the owner of the information, in most cases this should be the owner of the information. This will result in a Show Verb with the information requested for the Noun as the response." (Open Applications Group 2)
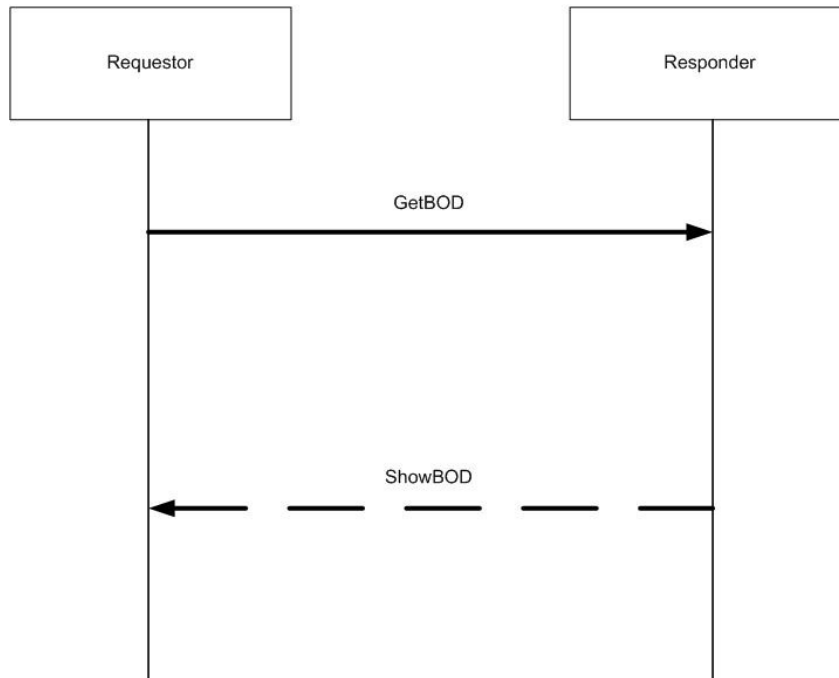
Get & Show

Figure 14 Get & Show Sequence Diagram
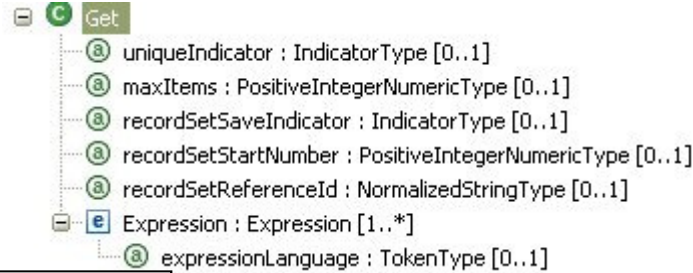
5.4.1. Get

OAGIS defines the Get verb as follows:

"The Get verb is to communicate to a business software component a request for an existing piece of information to be returned. The Get may be paired with most of the nouns defined in the OAGIS specification. The response to this request is the Show verb. The behavior of a BOD with a Get verb is quite predictable across most of the nouns it may be paired with. The Get is designed to retrieve a single piece of information by using that information's

primary retrieval field, or key field. The Get verb is not used to request several documents at once. The GetList verb is designed to achieve that purpose and will be covered in more detail later.

5.4.1.1. Selection Criteria

There are two types of selection capabilities for most BOD's that use the Get verb:

- The first selection capability is called Field-Based Selection. Within a Get-based Business Object Document, the first Data Type that occurs in a specific BOD structure is commonly used to provide the Field-Based Selection criteria. This is always defined within the specific BOD and is commonly the required fields for that specific Data type. The Field-Based Selection enables the requester to provide a value or values (in the case of multiple required Field Identifiers), in the required fields. Then the responding component uses those values to find and return the requested information to the originating business software component.

- The second type of selection capability for Get-based BODs is called Data Type Selection. Data Type selection enables the requester to identify which Data Types within the noun are requested to be returned in the response. The use of this capability is described for each corresponding Data Type for all BODs that use the Get verb. The Data Types are identified for retrieval within the Get instance of a BOD by including the name of the Data Type in the meta data but without any Field Identifiers or Segments identified within the Data Type. This will signify to the responding application that all of the data that corresponds to that Data Type is to be included in the response. If the Data Type is not requested, the Data Type identifier is not included in the Get request and this will signify to the responding component that the Data Type is not to be returned." (OAGIS 9.0 Meta.xsd)

Figure 15 Get Verb Attributes

GetVerb	Definition
uniqueIndicator	The purpose of the uniqueIndicator attribute on the Get verb is to indicate whether or not the request for the retrieval of information is unique.
maxItems	The purpose of the maxItems attribute on the Get verb is to indicate the maximum number of items to be returned in the retrieval process.
recordSetSaveIndicator	The purpose of the recordSetSaveIndicator attribute on the Get verb is to indicate that the record set being retrieved should be saved.
recordSetStartNumber	The purpose of the recordSetStartNumber attribute on the Get verb is to indicate the number of the record set where the retrieval process should begin.
recordSetReferenceId	The purpose of the recordSetReferenceId attribute on the Get verb is to indicate a reference number to identify the record set being retrieved.

Table 7 Get Verb Attributes Defined

5.4.2. Show

OAGIS defines the Show verb as follows:

"The Show verb is used when sending the information about a specific instance of a business document or entity. The Show verb may be used to respond to a Get request or it can be used in a publish scenario, where it pushes information to other applications based on a business event. Although BODs based on this verb do not commonly cause updates to occur, there may be times when the component receiving the Show decides to use the information it receives to update. This is entirely the decision of the receiving software component and is not forbidden. The behavior of the Show verb is quite straight forward with one exception. The Show response to any Get request needs to read the request carefully to ensure the response is returning the requested Data Types." (OAGIS 9.0 Meta.xsd)
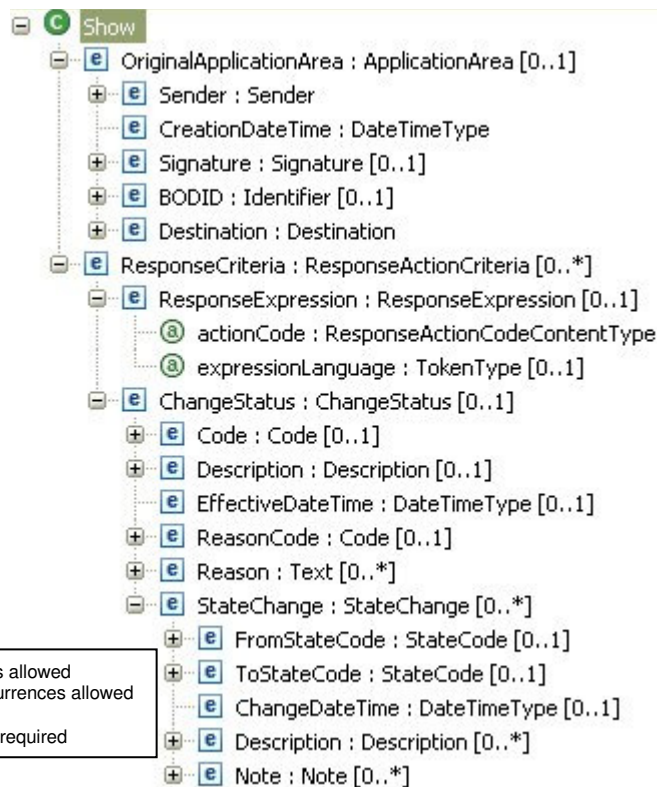

Figure 16 Show Verb Attributes

ShowVerb	Definition
OriginalApplicationArea	A copy of the ApplicationArea for the original BOD that was processed. Present either as additional reference information, or for use in identifying the BOD in situations where a BODReference is not known.
ResponseCriteria	ResponseCriteria identifies the content that is to be returned, given query success. In essence, the expression here has the effect of filtering the part(s) of the found element(s) that are to be returned. ReturnCriteria plays no role in the query itself. That is handled as a match against the request BOD's noun exemplar. ReturnCriteria allows the sender of the BOD to indicate which information (down to the field level) is requested to be returned, given that the query has been successful in matching the exemplar to existing nouns. That is, in a GetListPurchaseOrder, if one or more PurchaseOrders with a TotalPrice = $1M were found, ReturnCriteria tells the BOD recipient which parts of the PurchaseOrder should be populated with content when the response (ShowPurchaseOrder) is formulated. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 8 Show Verb Attributes Defined

5.5. SYNC & CONFIRM BOD

The following verb usage description has been defined by OAGIS:

"If an entity is changed in the system of record and other systems need to be made aware of the change use the Sync Verb. The Sync Verb is sent from the owner or system of record of the information.

The entity could change through user interface or by processing an incoming BOD message. It should be determined in advance who or what systems publish data and who does not.

In this manner the Sync acts as an event notification to any interested subscribers. If anybody is interested, they can update their own data based on this event notification." (Open Applications Group 2)
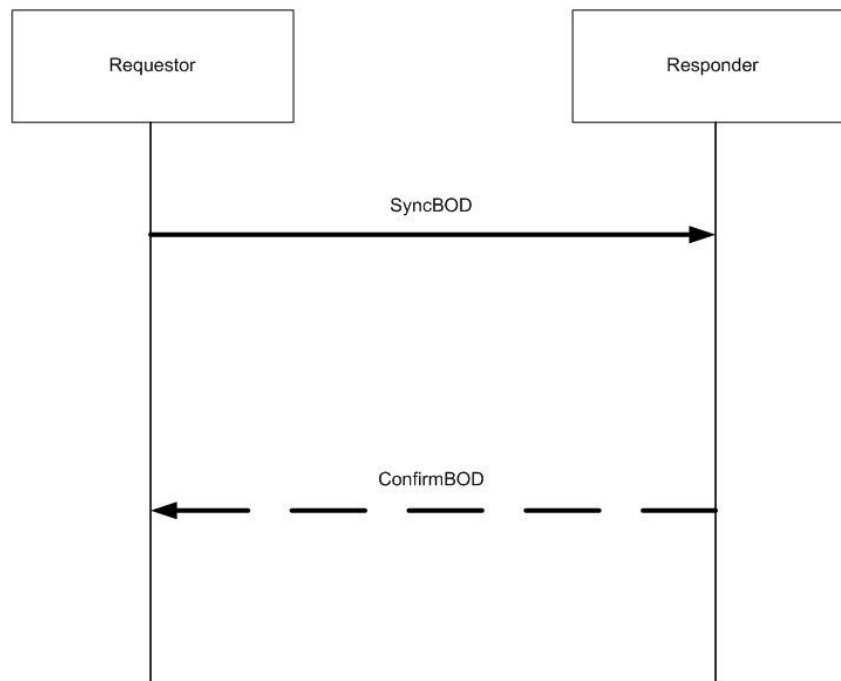
Sync & Confirm

Figure 17 Sync & Confirm Sequence Diagram

5.5.1. *Sync*

OAGIS defines the Sync as follows:

> "The Sync verb is used when the owner of the data is passing or publishing that information or change in information to other software components. This is to be used when the receiver of the SyncBOD does not own the data. This verb is commonly used when mass changes are necessary or when a publish and subscribe mechanism is used in the integration architecture. The purposes of this verb include application integrity and ease of data entry for the business user by enabling a single point of input." (OAGIS 9.0 Meta.xsd)
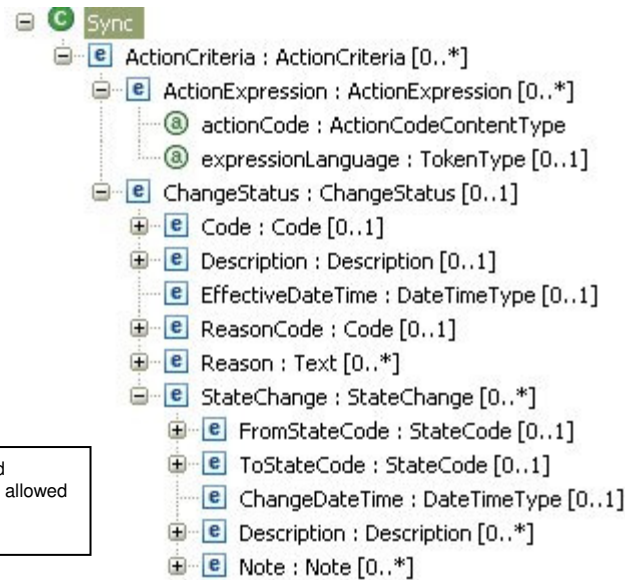
[0..1] = 1 optional occurrences allowed
[0..*] = 0 to many optional occurrences allowed
[1..1] = 1 occurrence required
[1..*] = 1 or more occurrences required

Figure 18 Sync Verb Attributes

SyncVerb	Definition
ActionCriteria	Identifies with the ActionExpression the level by carrying a value of the expressionLanguage (this is typically XPath) within the BOD that the actionCode is to be performed. The ChangeStatus communicates just that the ChangeStatus.

ActionExpression	ActionExpression identifies the content that is to be acted upon, given query success. In essence, the expression here has the effect of filtering the part(s) of the found element(s) that are to be acted upon. The expressionLanguage indicates the expression language being used. In order for the ReturnCriteria expression to be evaluable by the BOD recipient, the recipient must be capable of processing and interpreting the specified expression language. XPath is the default, due to its ubiquity among XML processing technologies.
ChangeStatus	ChangeStatus indicates any changes in the status of a Noun or Nouns. The ChangeStatus component may be used to indicate a reason and/or reason code for a particular change such as an "Addition of a User", " Deletion of a User", etc. as well as the effective date and time of the change.
StateChange	StateChange is used for managing the state of the Noun or Nouns. It can used to keep track of the condition of a particular process.

Table 9 Sync Verb Attributes Defined

5.5.2. Confirm

See 5.2.2 Confirm

5.6. CONFIRM

The ConfirmBOD is used to respond to a request to confirm the receipt of information by a receiving system. The ConfirmBOD indicates whether or not the receiving system was able to process the original message. In order to process the original message successfully, the receiving system must be able to successfully process every line of code necessary to complete the original message. This processing includes all units of work from parsing the message to processing the message through a backend legacy system.

The request for confirmation is set by the sending application in the verb field of the BOD. All STAR BODs require one Verb element that can be found in the DataArea of the BOD. The Verb indicates the action to be performed on the Noun.

All Verb elements contain a confirm attribute. The confirm attribute has three possible settings:

1. Always - The ConfirmBOD should always be sent.

2. Never - The ConfirmBOD should never be sent.

3. OnChange - The ConfirmBOD should be sent on change (i.e., only be sent when there are errors).

In order for a ConfirmBOD to be sent, the Sender of the original message must select one of the three confirm attribute values. If the Sender does not elect to receive a Confirm, a ConfirmBOD should not be sent. This is because the Sender, not expecting the ConfirmBOD, would have no way of handling the BOD. For simplicity, STAR suggests as a best practice that the OnChange attribute be used when requesting a ConfirmBOD.

5.6.1. Types of Errors Covered

The ConfirmBOD covers syntactical as well as business content-related errors that are not specific to a particular BOD. If an error code is defined in the ConfirmBOD as well as a response-type message (e.g., AcknowledgePartsOrder) it is recommended that the ConfirmBOD be used for implementation consistency. The ConfirmBOD may also be used to indicate that the receiver of the original message was not able to generate a successful response message. Example scenarios include the receiver is not able to properly validate the outbound message; the outbound message could not be properly compressed, etc. In scenarios such as these, the ConfirmBOD would be sent to the original sender indicating the failure on the part of the receiving system to generate a valid response. For the sender's reference, the ConfirmBOD would contain the original ApplicationArea of the original message sent.

6. STANDALONE BODS

STAR BODs are made up of data points from several individual xsd files, e.g. the Components.xsd, Fields.xsd, Codelist.xsd, OAGIS Verbs, etc. Some of these xsd files are added to the BOD through the use of include statements. These files are found in the STAR namespace. Others xsd files are added through the use of import statements. Files that are imported are found in alternative namespaces such as OAGI. In order to reduce the number of files associated with the processes of a single BOD, STAR has provided "Standalone" BODs.

A Standalone BOD is a consolidation of all relevant data points, i.e., components, fields, codelists, that are specific to that particular BOD from the Resource xsd files that are contained within the STAR namespace. As a result the Standalone BOD no longer has to "include" multiple Resource files. The Stanalone does not however consolidate data points from xsd files outside of the STAR namespace. Those files will still need to be imported and available for validation.

STAR recommends using the Developer when implementing more than one schema, and doing schema caching. If only a limited number of STAR BODs are being implemented, and ability to cache the schemas is not available, then implement the Standalone version of the STAR BOD.

If a noun-specific WSDL is being implemented the Developer BODs must be used. Otherwise, namespace conflicts for schema definitions will occur in the STAR namespace.

7. DOCUMENT IDENTIFICATION GROUP

The DocumentIdentificationGroup component is found in the Header of all STAR BODs. The purpose of the component is to provide a list of identifications related to a particular document.   These identifications may be used to uniquely identify a particular document.

It is currently the recommendation of STAR that the DocumentIdentification portion of the DocumentIdentificationGroup be used as the Unique ID for the document, preferably globally unique. The AlternateDocumentIdentification portion of the component should be used for system specific IDs.   These are however only recommendations and each implementation may require a different implementation of the component.

Figure 19 Document Identification Group

7.1.   DOCUMENTID

The DocumentIdentification component contains a DocumentID field. This identifier is to be used as the Unique ID for the document. The DocumentID field as several optional scheme attributes as shown in **Figure 20 DocumentID Scheme Attributes**. These attributes can be used to identify the entity that assigned the DocumentID. For example:

- schemeName: OrderID

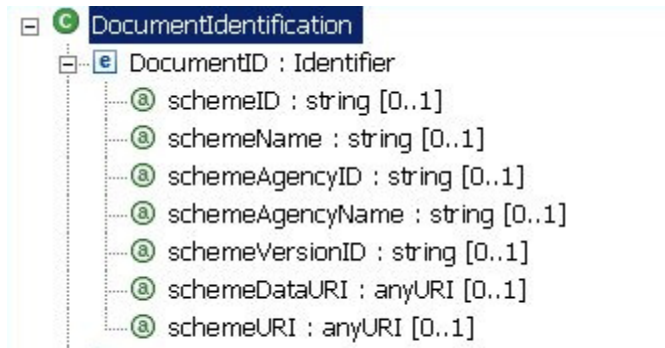- schemeAgencyName: Nav International

Figure 20 DocumentID Scheme Attributes

8. PARTY IDENTIFICATION

A majority of Parties defined within STAR, i.e., BillToParty, ShipToParty, etc. are based on a single Party component. This Party component includes a field call **PartyID** and a component call **AlternatePartyDocument**.

The PartyID should be used as a unique identifier for a particular party. This may be a globally unique ID or it may be a system ID.

The AlternatePartyDocument component should be used supplying related documents for a Party such as a driver's license, social security number, etc. This component should not be used for system IDs.
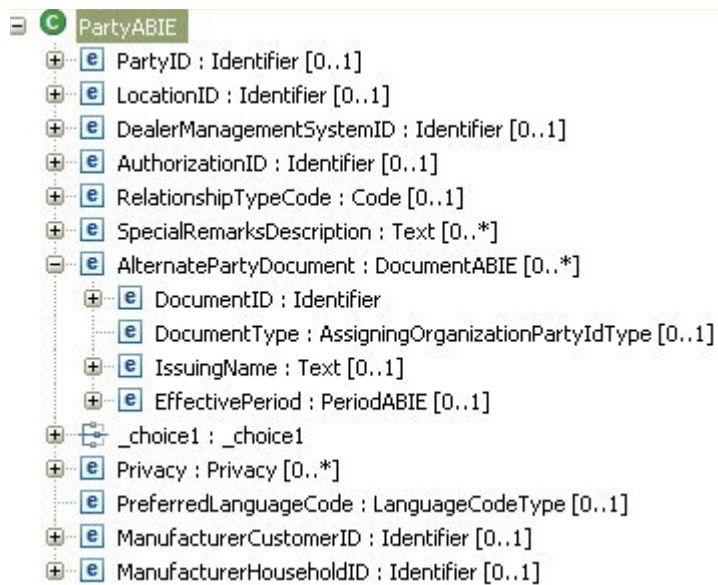

Figure 21 Party Identification Elements

9. WORKS CITED

Open Applications Group, Inc. "Using OAGIS Verbs". 2006.

Open Applications Group, Inc. "Meta.xsd". OAGIS 9.0. 2005.