**Schema/Document File**: BatchML-V0600-BatchInformation.xsd

**Type/Element/Attribute**: BatchValueType

**Classification**:

**[ ]** me - minor ediorial **[X]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

The extension type of BatchValueType is wrong. It is “Value” and should be “BatchValue”:

<!-- Complex Type with Extension -->

<xsd:complexType name = "BatchValueType" mixed = "true">

<xsd:sequence>

<xsd:element name = "ValueString" type = "ValueStringType"

minOccurs = "1" maxOccurs = "unbounded"/>

<xsd:element name = "DataInterpretation" type = "DataInterpretationType"/>

<xsd:element name = "DataType" type = "DataTypeType" />

<xsd:element name = "UnitOfMeasure" type = "UnitOfMeasureType" />

<xsd:element name = "EnumerationSetID" type = "EnumerationSetIDType"

minOccurs = "0" maxOccurs = "unbounded"/>

<xsd:group ref = "Extended:**Value**" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

**Resolution**:

Change the extended reference to “BatchValue”.

…

<xsd:group ref = "Extended:**BatchValue**" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

**Submitted by email by:**

Ross Wilcox 10/3/2013

Principal Consultant

DuPont Engineering

**Schema/Document File**: BatchML-V0600-BatchInformation.doc

**Type/Element/Attribute**: NA

**Classification**:

**[X]** me - minor ediorial **[ ]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

The “LinkType documentation contains an error, the terms SequenceDivergent and SequenceConvergent should be SerialDivergent and SerialConvergent:

|  |  |
| --- | --- |
| LinkType  ***LinkTypeType*** | … Standard enumerations are:  "ControlLink", "TransferLink", "SynchronizationLink", "ParallelDivergent", "ParallelConvergent", "**SequenceDivergent**", “**SequenceConvergent**", and "Other"   * … * **SequenceDivergent** 🡪 The link is part of a diverging selection structure (e.g. one of multiple links from one step to multiple transitions, defining a selection branch). * **SequenceConvergent** 🡪 The link is part of a converging selection structure (e.g. one of multiple links from steps to a single transition that defines the end of selection branches).   If “Other” then the type is an application specific extension and the value is defined in the attribute “OtherValue”. |

**Resolution**:

Documentation corrected:

|  |  |
| --- | --- |
| LinkType  ***LinkTypeType*** | … Standard enumerations are:  "**ControlLink**", "**TransferLink**", "**SynchronizationLink**", "**ParallelDivergent**", "**ParallelConvergent**", " **SerialDivergent** ", “**SerialConvergent** ", and "**Other**"   * … * SerialDivergent 🡪 The link is part of a diverging selection structure (e.g. one of multiple links from one step to multiple transitions, defining a selection branch). * SerialConvergent 🡪 The link is part of a converging selection structure (e.g. one of multiple links from steps to a single transition that defines the end of selection branches).   If “Other” then the type is an application specific extension and the value is defined in the attribute “OtherValue”. |

**Submitted by email by:**

Gabor Hodi

**Schema/Document File**: BatchML-V0600-GeneralRecipe.docx

**Type/Element/Attribute**: NA

**Classification**:

**[ X]** me - minor ediorial **[ ]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

The figure for GRecipeMaterialsType is wrong. It is showing the GRecipeInformationType

**Resolution**:

Figure changed to the correct figure:

|  |  |
| --- | --- |
| ***GRecipeMaterialsType***  Materials  ProcessInputs  ProcessOutputs  ProcessIntermediates | A *GRecipeMaterialsType* is used for Process Inputs, Process Outputs, and Materials (in a Process Element) to define a set of materials.  Each *GRecipeMaterialType* element contains an optional identification, a definition of the use of the materials (*MaterialsTypeType*), and a set of Material (*GRecipeMaterialType*) definitions. |

**Submitted by email by:**

Daniel Brandl, NNE Pharmaplan

**Schema/Document File**: B2MML-V0600-Common.xsd

**Type/Element/Attribute**: In EquipmentAssetMappingType

**Classification**:

**[ ]** me - minor ediorial **[X]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

In EquipmentAssetMappingType, the StartTime and EndTime are of generic type DateTimeType and not specialized StartTimeType or EndTimeType from common:

<xsd:complexType name="EquipmentAssetMappingType">

<xsd:sequence>

<xsd:element name="EquipmentID" type="EquipmentIDType"/>

<xsd:element name="PhysicalAssetID" type="PhysicalAssetIDType"/>

<xsd:element name="StartTime" type="**DateTimeType**" minOccurs="0"/>

<xsd:element name="EndTime" type="**DateTimeType**" minOccurs="0"/>

<xsd:group ref="Extended:EquipmentAssetMapping" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

**Resolution**:

Types will be changed to the StartTimeType and End.

<xsd:complexType name="EquipmentAssetMappingType">

<xsd:sequence>

<xsd:element name="EquipmentID" type="EquipmentIDType"/>

<xsd:element name="PhysicalAssetID" type="PhysicalAssetIDType"/>

<xsd:element name="StartTime" type="**StartTimeType**" minOccurs="0"/>

<xsd:element name="EndTime" type="**EndTimeType**" minOccurs="0"/>

<xsd:group ref="Extended:EquipmentAssetMapping" minOccurs="0" maxOccurs="1"/>

</xsd:sequence>

</xsd:complexType>

**Submitted by email by:**

Krister Thelin, Volvo

**Schema/Document File**: General Questions

**QUESTION:**

ConfidenceFactor is of type IdentifierType, should it be a decimal/numeric value?

  <xsd:complexType name="ConfidenceFactorType">  
        <xsd:simpleContent>  
            <xsd:restriction base="IdentifierType"/>  
        </xsd:simpleContent>  
    </xsd:complexType>

**RESPONSE:**

The ISA-95 Part 2, 2010 edition, Table 81 shows that this may not always be a number. It may also be a string such as “Medium”. B2MML implemented this using the IdentifierType to allow non-numerical confidence factors.

**QUESTION**:

EquipmentUse, PersonnelUse, PhysicalAssetUse are all opposed to MaterialUse, without a set enumeration. Is there a plan to introduce enumerated values for the others?

**RESPONSE:**

The ISA-95 Part 2, 2010 edition, in section 6.2.12, below table 67 requires the enumerations for MaterialUse, it does not provide any for the other types. Therefore B2MML only provides them for MaterialUse.

**QUESTION**:

In EquipmentType, should both HierarchyScope and EquipmentLevel be of type HierarchyScopeType? Or should the EquipmentLevel  actually be EquipmentElementLevel?

<xsd:complexType name = "EquipmentType">  
    <xsd:sequence>  
      <xsd:element name = "ID"                  type = "IdentifierType"/>  
      <xsd:element name = "Description"         type = "DescriptionType"          
                                                minOccurs = "0" maxOccurs = "unbounded"/>  
      <!-- Location ELEMENT IS DEPRECATED and may be removed in a future release, use   
 HierarchyScope instead -->  
      <xsd:element name = "Location"            type="LocationType"   
                                             minOccurs="0"/>   
      <xsd:element name = "HierarchyScope"      type = "HierarchyScopeType"     
                                                minOccurs = "0" />  
      <xsd:element name = "EquipmentLevel"      type = "HierarchyScopeType"     
                                                minOccurs = "0" />  
      <xsd:element name = "EquipmentAssetMapping" type = "EquipmentAssetMappingType"     
                                                minOccurs = "0" maxOccurs = "unbounded"/>  
      <xsd:element name = "EquipmentProperty"   type = "EquipmentPropertyType"    
                                                minOccurs = "0" maxOccurs = "unbounded"/>  
      <xsd:element name = "Equipment"           type = "EquipmentType"    
                                                minOccurs = "0" maxOccurs = "unbounded"/>  
      <xsd:element name = "EquipmentClassID"    type = "EquipmentClassIDType"     
                                                minOccurs = "0" maxOccurs = "unbounded"/>  
      <xsd:element name = "EquipmentCapabilityTestSpecificationID"  
                                               type = "EquipmentCapabilityTestSpecificationIDType"  
                                                minOccurs = "0" maxOccurs = "unbounded"/>  
      <xsd:group   ref  = "Extended:Equipment"  minOccurs = "0" maxOccurs = "1"/>  
    </xsd:sequence>  
  </xsd:complexType>

**RESPONSE**:

HierarchyScopeType was used for both elements since both types would have had identical definitions. Doing this indicates that they have a similar structure. Sharing the same type was also done for a few other elements.

**QUESTION**:

The data areas is inconsistently named. It seems more correct to with naming like SyncEquipmentDataAreaType versus existing SyncEquipmentTypeDataArea. Compared to OpMaterialCapabilityType and its OpMaterialCapabilityPropertyType.  
Today a lot of data and text processing is done by tools and templates. Having a consistent naming really helps there.

**RESPONSE**:

In all the transaction elements the DataArea elements are consistently defined, but they are not created based upon a type.

Different tools handle code generation differently, in the past we spent time chasing different tools, now we ask people to contribute to the B2MML-BatchML-V0600-CodeGeneration.pdf document to help others understand changes required in the schemas so they work better with a specific tool.

**QUESTION**:

There are some unused complexTypes not being used inside the core components. Re they there for backwards compability with Any element and/or with custom extensions? Thinking about AmountType and BinaryObjectType.

**RESPONSE**:

The unused complexTypes were included for consistency with UN/CEFACT and to allow people to use them with extensions.

**Schema/Document File**: B2MML V0600 Work Definition

**Type/Element/Attribute**: B2MML-V0600-WorkDefinition.xsd

**Classification**:

**[ ]** me - minor ediorial **[X]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

The ISA 95 standard shows a loose association between the Work Master and Operations Definition package. This relations is labeled as “may have a reference to”. The text of the standard says “A work definition may have a reference to an operations definition. In this situation the work definition defines the detailed steps needed to accomplish all or part of the operation. ”

The current B2MML implementation only has a simple reference from the WorkDefinition to OperationsDefinition, through the OperationsDefinitionID element.

B2MML V0600 is too restrictive to represent:

1. The fact that a WorkMaster may be used in multiple OperationsDefinitions.
2. The WorkMaster may only accomplish part of the operation, so it should reference the elements making up the Operations Definition, the Operations Segment.

Extending these allows the exchange of information that relates a WorkMaster to one or more OperationsDefinitions or one or more OperationsSegments, allowing for full mapping of how work is performed to the request for work.

**Current Wording or Schema Definition**:

<xsd:group name = "WorkDefinitionType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType" />

<xsd:element name = "Version" type = "VersionType"

minOccurs = "0"/>

…

<xsd:element name = "PublishedDate" type = "PublishedDateType"

minOccurs = "0"/>

**<xsd:element name = "OperationsDefinitionID"**

**type = "OperationsDefinitionIDType"**

**minOccurs = "0"/>**

<xsd:element name = "Parameter" type = "ParameterType"

minOccurs = "0" maxOccurs = “unbounded"/>

…

</xsd:sequence>

</xsd:group>

**Proposed Wording or Schema Definition**:

<xsd:group name = "WorkDefinitionType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType" />

<xsd:element name = "Version" type = "VersionType"

minOccurs = "0"/>

…

<xsd:element name = "OperationsDefinitionID"

type = "OperationsDefinitionIDType"

minOccurs = "0" **maxOccurs = "unbounded"**/>

<**xsd:element name = "OperationsSegmentID"**

**type = "OperationsSegmentIDType"**

**minOccurs = "0" maxOccurs = "unbounded"/>**

<xsd:element name = "Parameter" type = "ParameterType"

minOccurs = "0" maxOccurs = “unbounded"/>

…

</xsd:sequence>

</xsd:group>

This should also include the following text in the documentation:

A WorkDefinition may reference multiple different OperationsDefinition, or multiple parts of OperationsDefinitions.

* If the WorkDefinition references the entire OperationsDefinition, the OperationsDefinitionID contains the OperationsDefinition ID.
* It the WorkDefinition references part of an OperationsDefinition and the OperationsSegment IDs are not unique across all OperationsDefinitions, then the OperationsSegmentID should contain the entire ID path to the OperationsSegment.
  + For Example: “R123/020/010” for the Operations Segment 010, within Operations Segment 020 within Operations Request R123.

**Name, Company, Contact Information**:

Dennis Brandl, BR&L Consulting, Inc. [dnbrandl@brlconsulting.com](mailto:dnbrandl@brlconsulting.com)

**Schema/Document File**: B2MML-V0600-OperationsSchedule.xsd

**Type/Element/Attribute**: OperationsRequestType, OpSegmentRequirementType

**Classification**:

**[ ]** me - minor ediorial **[X]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

The ISA 95 standard shows a loose association between the OperationsSchedule and Operations Definition package. This relations is labeled as “may correspond to”.

The current B2MML implementation only has a simple reference from the OperationsRequest and the SegmentRequest to OperationsDefinition, through the OperationsDefinitionID element.

B2MML V0600 too restrictive to represent the Operations Segments that make up the Operations Definition.

An Operations Request and Segment Requirement should be able to reference either an entire Operations Definition or an Operations Segment within an Operations Definition.

**Current Wording or Schema Definition**:

<xsd:complexType name = "OperationsRequestType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

… minOccurs = "0"/>

<xsd:element name = "**OperationsDefinitionID**" type = "OperationsDefinitionIDType"

minOccurs = "0"/>

<xsd:element name = "RequestState" type = "RequestStateType"

minOccurs = "0"/>

…

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name = "OpSegmentRequirementType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

<xsd:element name = "**OperationsDefinitionID**" type = "OperationsDefinitionIDType"

minOccurs = "0"/>

<xsd:element name = "SegmentState" type = "RequestStateType"

minOccurs = "0"/>

…

</xsd:sequence>

</xsd:complexType>

**Proposed Wording or Schema Definition**:

<xsd:complexType name = "OperationsRequestType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

… minOccurs = "0"/>

<xsd:element name = "OperationsDefinitionID" type = "OperationsDefinitionIDType"

minOccurs = "0" />

**<xsd:element name = "OperationsSegmentID" type = "OperationsSegmentIDType"**

**minOccurs = "0" />**

<xsd:element name = "RequestState" type = "RequestStateType"

minOccurs = "0"/>

…

</xsd:sequence>

</xsd:complexType>

<xsd:complexType name = "OpSegmentRequirementType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

<xsd:element name = "OperationsDefinitionID" type = "OperationsDefinitionIDType"

minOccurs = "0"/>

**<xsd:element name = "OperationsSegmentID" type = "OperationsSegmentIDType"**

**minOccurs = "0" />**

<xsd:element name = "SegmentState" type = "RequestStateType"

minOccurs = "0"/>

…

</xsd:sequence>

</xsd:complexType>

This should also include the following text in the documentation:

An OperationsRequest may reference a single OperationsDefinition, or part of an OperationsDefinitions.

* If the OperationsRequest references the entire OperationsDefinition, the OperationsDefinitionID contains the OperationsDefinition ID.
* It the OperationsRequest references part of an OperationsDefinition and the OperationsSegment IDs are not unique across all OperationsDefinitions, then the OperationsSegmentID should contain the entire ID path to the OperationsSegment.
  + For Example: “R123/020/010” for the Operations Segment 010, within Operations Segment 020 within Operations Request R123.

**Name, Company, Contact Information**:

Dennis Brandl, BR&L Consulting, Inc. [dnbrandl@brlconsulting.com](mailto:dnbrandl@brlconsulting.com)

**Schema/Document File**: B2MML-V0600-OperationsPerformance.xsd

B2MML-V0600-OperationsPerformanceTypes.xsd

**Type/Element/Attribute**: OperationsResponseType, OpSegmentResponseType

**Classification**:

**[ ]** me - minor ediorial **[X]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

The ISA 95 standard shows a loose association between the OperationsPerformance and Operations Definition package. This relations is labeled as “may correspond to”.

The current B2MML implementation only has a simple reference from the OperationsResponse and the SegmentRequest to ProcessSegment, although other text states that the SegmentResponse may be related to part of an OperationsRequest (An OperationsSegment), through the OperationsDefinitionID element.

B2MML V0600 is too restrictive to represent the Operations Segments that make up the Operations Definition.

An Operations Response and Segment Response should be able to reference either an entire Operations Definition or an Operations Segment within an Operations Definition.

**Current Wording or Schema Definition**:

B2MML-V0600-OperationsPerformance.xsd

<xsd:complexType name = "OperationsResponseType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

<xsd:element name = "**OperationsDefinitionID**" type = "**OperationsDefinitionIDType**"

minOccurs = "0" maxOccurs = "unbounded"/>

<xsd:element name = "ResponseState" type = "ResponseStateType"

minOccurs = "0"/>

…

</xsd:sequence>

</xsd:complexType>

B2MML-V0600-OperationsPerformanceTypes.xsd

<xsd:complexType name = "OpSegmentResponseType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

<xsd:element name = "**OperationsDefinitionID**" type = "**OperationsDefinitionIDType**"

minOccurs = "0" maxOccurs = "unbounded"/>

<xsd:element name = "SegmentState" type = "ResponseStateType"

minOccurs = "0"/>

…

</xsd:complexType>

**Proposed Wording or Schema Definition**:

B2MML-V0600-OperationsPerformance.xsd

<xsd:complexType name = "OperationsResponseType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

<xsd:element name = "OperationsDefinitionID" type = "OperationsDefinitionIDType"

minOccurs = "0" maxOccurs = "unbounded"/>

**<xsd:element name = "OperationsSegmentID" type = "OperationsSegmentIDType"**

**minOccurs = "0" maxOccurs = "unbounded"/>**

<xsd:element name = "ResponseState" type = "ResponseStateType"

minOccurs = "0"/>

…

</xsd:sequence>

</xsd:complexType>

B2MML-V0600-OperationsPerformanceTypes.xsd

<xsd:complexType name = "OpSegmentResponseType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

<xsd:element name = "OperationsDefinitionID" type = "OperationsDefinitionIDType"

minOccurs = "0" maxOccurs = "unbounded"/>

**<xsd:element name = "OperationsSegmentID" type = "OperationsSegmentIDType"**

**minOccurs = "0" maxOccurs = "unbounded"/>**

<xsd:element name = "SegmentState" type = "ResponseStateType"

minOccurs = "0"/>

…

</xsd:complexType>

This should also include the following text in the documentation:

An OperationsResponse or a SegmentResponse may reference a single OperationsDefinition, or part of an OperationsDefinitions.

* If it references the entire OperationsDefinition, the OperationsDefinitionID contains the OperationsDefinition ID.
* It it references part of an OperationsDefinition and the OperationsSegment IDs are not unique across all OperationsDefinitions, then the OperationsSegmentID should contain the entire ID path to the OperationsSegment.
  + For Example: “R123/020/010” for the Operations Segment 010, within Operations Segment 020 within Operations Request R123.

**Name, Company, Contact Information**:

Dennis Brandl, BR&L Consulting, Inc. [dnbrandl@brlconsulting.com](mailto:dnbrandl@brlconsulting.com)

**Schema/Document File**: B2MML-V0600-WorkSchedule.xsd

B2MML-V0600-WorkPerformance.xsd

**Type/Element/Attribute**: JobOrderType, JobResponseType

**Classification**:

**[ ]** me - minor ediorial **[X]** mt - minor technical

**[ ]** ME - Major Editorial **[ ]** MT - Major Technical

**Comment**:

The ISA 95 standard defines the activity of Production Tracking to include the tracking of Job Responses for a Job Order back to the originating Operations Request (or Segment Requirement). However, there is no way for different applications to know the relationship between Operations Requests and Job Orders in order to actually perform the tracking activity.

Job orders should have an optional definition of the associated Operations Requests or Segment Requirements.

Job Responses should have an optional definition of the associated Operations Requests or Segment Requirements

This change does not affect existing B2MML implementations, but does allow the tracking activity to be used with a scheduling activity, without the exchange of information that is otherwise not available in a B2MML format.

**Current Wording or Schema Definition**:

B2MML-V0600-WorkSchedule.xsd

<xsd:complexType name = "JobOrderType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

<xsd:group ref = "Extended:JobOrder" minOccurs = "0" maxOccurs = "1"/>

</xsd:sequence>

</xsd:complexType>

B2MML-V0600-WorkPerformance.xsd

<xsd:complexType name = "JobResponseType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

</xsd:sequence>

</xsd:complexType>

**Proposed Wording or Schema Definition**:

B2MML-V0600-WorkSchedule.xsd

<xsd:complexType name = "JobOrderType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

**<xsd:element name = "OperationsRequestID" type = "OperationsRequestIDType"**

**minOccurs = "0" maxOccurs = "unbounded"/>**

**<xsd:element name = "SegmentRequirementID" type = "SegmentRequirementIDType"**

**minOccurs = "0" maxOccurs = "unbounded"/>**

…

<xsd:group ref = "Extended:JobOrder" minOccurs = "0" maxOccurs = "1"/>

</xsd:sequence>

</xsd:complexType>

B2MML-V0600-WorkPerformance.xsd

<xsd:complexType name = "JobResponseType">

<xsd:sequence>

<xsd:element name = "ID" type = "IdentifierType"

minOccurs = "0"/>

…

**<xsd:element name = "OperationsRequestID" type = "OperationsRequestIDType"**

**minOccurs = "0" maxOccurs = "unbounded"/>**

**<xsd:element name = "SegmentRequirementID" type = "SegmentRequirementIDType"**

**minOccurs = "0" maxOccurs = "unbounded"/>**

…

</xsd:sequence>

</xsd:complexType>

This should also include the following text in the documentation:

A JobOrderType or a JobResponseType may reference one or more OperationsRequests, or parts of an OperationsRequest.

* If it references the entire OperationsRequest, the OperationsRequestID contains the OperationsRequest ID.
* It it references part of an OperationsRequest, then the SegmentRequirementID, and the SegmentRequirementIDs are not unique across all OperationsRequests, then the SegmentRequirementID should contain the entire ID path to the SegmentRequirementID.
  + For Example: “SCH123/020/010” for the:
    - Segment Requirement 010,
    - within Segment Requirement 020
    - within Operations Request SCH123.

**Name, Company, Contact Information**:

Dennis Brandl, BR&L Consulting, Inc. [dnbrandl@brlconsulting.com](mailto:dnbrandl@brlconsulting.com)