

Business To Manufacturing   
Markup Language

Work Record

Version 0700

August 7, 2016

B2MML - Work Record

IMPORTANT: While the information, data, and standards provided in this publication were developed and are presented in good faith in accordance with a reasonable process that was subject to intellectual property and antitrust policies to benefit the industry as a whole, the publication is provided “as is” for information and guidance only, and there is no representation or warranty of any type or kind, including but not limited to warranties of merchantability or fitness for a particular purpose, and no warranty that use of the information, data, or standards will not infringe patent, copyright, trademark, trade secret, or other intellectual property rights of any party.

Copyright © 2016 MESA International

All Rights Reserved. http://www.mesa.org

This MESA Work (including specifications, documents, software, and related items) referred to as the Business To Manufacturing Markup Language (B2MML) is provided by the copyright holders under the following license.

Permission to use, copy, modify, or redistribute this Work and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted provided MESA International is acknowledged as the originator of this Work using the following statement:

"The Business To Manufacturing Markup Language (B2MML) is used courtesy of MESA International."

In no event shall MESA International, its members, or any third party be liable for any costs, expenses, losses, damages or injuries incurred by use of the Work or as a result of this agreement.

Material from ANSI/ISA-88 and ANSI/ISA-95 series of standards used with permission of ISA - The Instrumentation, Systems, and Automation Society, www.isa.org

Table of Contents

­

Change history 3

1 Schema Scope 4

1.1 Referenced Schemas 4

1.2 Key Use Assumptions 4

1.3 Key Information Assumptions 4

1.4 Common Data Types 4

1.5 Included Schemas 4

1.6 Core Components 5

2 Schema Organization 5

2.1 WorkRecord Element 5

2.2 Type Names 5

2.3 User Element Extensibility 5

3 WORK record Models 6

3.1 Work record 6

3.2 Records with references to other records 9

3.3 Event model 9

3.4 Sample model 10

3.5 Data set model 10

3.6 Operations elements 11

3.7 Production elements 11

3.8 Work elements 11

4 Element Definitions 12

4.1 Transactions 12

4.2 WorkRecordEntryType Group 13

4.3 Work Record Element 15

4.4 Data Elements 28

5 Diagram Convention 36

# Change history

|  |  |  |  |
| --- | --- | --- | --- |
| **Change** | **Date** | **Person** | **Description** |
| V0700 | Aug 2016 | D. Brandl | Initial Version |

# Schema Scope

This document defines the information about the definition of work report information that may be exchanged by manufacturing operations management systems. This information is based on the data models and attributes defined in the ANSI/ISA 95.00.04 Enterprise/Control System Integration standard and the ANSI/ISA 88.00.04 Batch Production Record Information. Contact ISA (The Instrumentation, System, and Automation Society) for copies of the standard. Additional information on the standard is available at [www.isa.org](http://www.isa.org) .

## Referenced Schemas

This document provides addresses the contents of the following MESA XML schema:

**B2MML-WorkRecord.xsd**

## Key Use Assumptions

The schemas define exchanged information and do not define the use of the information or encapsulation of the information in any defining transactions. These schemas are intended to be used to create XML documents used to exchange batch data as well as serve as the basis for corporate, system or application specific schemas that may be derived from the B2MML schemas.

## Key Information Assumptions

The schemas define simple and complex types and elements for recipe, equipment and batch list data commonly found in batch applications. A set of data models is presented for recipes, equipment and batch lists. Each model also illustrates the equivalent top-level XML elements that correspond to top-level objects identified in the ANSI/ISA-88 standard. The details of the schema element and attribute definitions are contained in later sections of this document.

## Common Data Types

The BatchML BatchInformation schema used the B2MML Common schema to pick up common data types.

See the documentation for the Common Types in the file:

**B2MML-Common.doc**

## Included Schemas

The following schemas are included as part of the Work Record schema:

**BatchML-BatchProductionRecord.xsd**

**B2MML-OperationsDefinition.xsd**

**B2MML-OperationsPerformance.xsd**

**B2MML-OperationsSchedule.xsd**

**B2MML-WorkAlert.xsd**

**B2MML-WorkCalendar.xsd**

**B2MML-WorkDefinition.xsd**

**B2MML-WorkPerformance.xsd**

**B2MML-WorkSchedule.xsd**

## Core Components

The B2MML-WorkRecord schema uses the B2MML Core Component schemas.

The base types for most elements are derived from core component types that are compatible with the UN/CEFACT core component types. The UN/CEFACT core component types are a common set of types that define specific terms with semantic meaning (e.g. the meaning of a quantity, currency, amount, identifier …). The UN/CEFACT core components were defined in a Core Components Technical Specification (CCTS) developed by the ebXML project now organized by UN/CEFACT and ISO TC 154.

The core components are defined in the schema file:

**B2MML-CoreComponents.xsd**

# Schema Organization

## WorkRecord Element

The B2MML root element is WorkRecord.

## Type Names

The XML schema uses a model that defines simple and complex data types for each element. The data types all follow the convention of a suffix of “Type” added to the element name.

Schema definition:

<xsd:element name = "**ApprovalDate**" type = "**ApprovalDateType** ==

<xsd:simpleType name="**ApprovalDateType**">

<xsd:restriction base="xsd:dateTime">

</xsd:restriction>

</xsd:simpleType>

## User Element Extensibility

In order to make the schemas more useful, they include the ability for elements to be extended. The extended elements are not defined in this standard and should not be considered understandable between applications without prior agreement.

See the definition of user extensions in:

**B2MML-Extensions.doc**

# WORK record Models

The exchanged information is derived from the UML models below.

NOTE: This version has used the IEC 62264.2 and IEC 62264.4 definitions definition for the ISA 95 data elements, the ANSI/ISA 88.00.01, ANSI/ISA 88.00.02 definitions, and definitions in ANSI/ISA 88.04.

## Work record

Work records are made up of multiple sub-records acting as container objects. These are represented as XML elements. Each sub-record is a collection of similarly typed individual records.

A Work Record may also contain another Work Record and zero or more Batch Production Records.

**Error! Reference source not found.** is the UML model from IEC/ISO 62264.4. is the UML model derived from ANSI/ISA-88.04 and ANSI/ISA95.02-2010. Each of the sub-record container elements contains a collection of zero or more individual records. These are modeled in B2MML as XML elements within the sub-record elements.

All of the individual records are specialized types derived from a Work Record Entry abstract type. This is modeled in B2MML using the “group” ***WorkRecordEntryType*** element that is included as the initial element in each of the individual records.



Figure - Work Record Structure



Figure - Batch production record structure

## Records with references to other records

The following four individual records may reference another records; comments, personnel identification manifest, qualification manifest, and change history. These are modeled in UML as in . These are modeled as a ***RecordReference*** element, which is defined as a Core Component ***IdentifierType***. This should contain the EntryID value of the referenced value. Illustrates the use of references.

Example 1: The EntryID contains an XPath specification string that identifies the element in the XML record.

Example 2: The EntryID contains a unique number that for each element.



Figure - Record with references to other records

## Event model

An event is a discrete occurrence in time. There are different types of event objects and each event is identified by an event type and subtype. Some event objects may be associated with other events, these are modeled as a ***RecordReference*** element, which is defined as a Core Component ***IdentifierType***. This should contain an XPath specification string that identifies the element in the XML record.

Events contain user defined records, may contain alarm event information and a set of event associations. User defined attributes are also derived from the Work Record Entry type. is the simplified UML model.



Figure - Event model

## Sample model

Samples contain Sample Tests, and Sample Tests contain Sample Test Results. Sample Tests and Sample Test Results are also derived from the Work Record Entry type. is a simplified UML model.



Figure - Sample model

## Data set model

Data sets are used to represent multivariable data sets or time based data sets. The sets contain ordered data sets. is the UML model from ANSI/ISA-88.04. The ordered data set is modeled using a “choice” element as either:

1. A Delimited data block in which the data sets are recorded in a single delimited string, with the delimiting characters defined in the ***DelimitedDataBlockType***.
2. As a set of ***OrderedDataType*** elements. Each OrderedDataType contains an optional time element and a set of data elements.



Figure - Data sets

## Operations elements

The production request, production response, and product definition elements are defined in the IEC/ISO 62264.2 standards. The Work Record schema uses the element definitions defined in:

**B2MML-OperationsDefinition.xsd**

**B2MML-OperationsPerformance.xsd**

**B2MML-OperationsSchedule.xsd**

## Production elements

The production request, production response, and product definition elements are defined in the ANSI/ISA-95.02 standard. The Work Record schema uses the element definitions defined in:

**B2MML-ProductDefinition.xsd**

**B2MML-ProductionPerformance.xsd**

**B2MML-ProductionSchedule.xsd**

## Work elements

The production request, production response, and product definition elements are defined in the IEC/ISO 62264.4 standard. The Work Record schema uses the element definitions defined in:

**B2MML-WorkAlert.xsd**

**B2MML-WorkCalendar.xsd**

**B2MML-WorkCapability.xsd**

**B2MML-WorkDefinition.xsd**

**B2MML-WorkPerformance.xsd**

**B2MML-WorkSchedule.xsd**

# Element Definitions

## Transactions

The following top level elements for transactions are defined:

|  |  |
| --- | --- |
| **Transaction Element** | **Description** |
| **GetWorkRecord** | Defines a message used to implement a GET action on a Work Record.   * **Work Record ID specified**: Defines a request that the receiver is to return, in a SHOW message, all attributes and contained elements of the Work Record that matches the ID for the specified equipment scope. * **Wildcard Work Record ID specified**: Defines a request that the receiver is to return, in a SHOW message, all attributes and contained elements of all Work Records that match the ID wildcard for the specified equipment scope. * **No Work Record ID Specified**: Apply the additional constraints. * One or more of the following constraints may apply:   + **CreationDate specified**: Defines a request that the receiver is to return, in a SHOW message, Work Records with the specified creation date for the specified equipment scope.   + **WorkRecordSpec specified**: Defines a request that the receiver is to return, in a SHOW message, Work Records created with the WorkRecordSpec for the specified equipment scope.   + **LastChangedDate specified**: Defines a request that the receiver is to return, in a SHOW message, Work Records with the specified LastChangedDate for the specified equipment scope.   + **RecordStatus specified**: Defines a request that the receiver is to return, in a SHOW message, Work Records with the specified RecordStatus for the specified equipment scope.   + **Version specified**: Defines a request that the receiver is to return, in a SHOW message, Work Records with the specified Version for the specified equipment scope. |
| **ShowWorkRecord** | Response from a **GetWorkRecord** transaction. |
| **ProcessWorkRecord** | Defines a request that the receiver is to add a new Work Record. Any assigned IDs in the Work Record are returned in the **AcknowledgeWorkRecord** message. |
| **AcknowledgeWorkRecord** | Response to a **ProcessWorkRecord** transaction. |
| **ChangeWorkRecord** | Defines a request to change information in a Work Record, replacing the existing information with the information in the transaction message. |
| **RespondWorkRecord** | Response to the **ChangeWorkRecord** transaction. |
| **CancelWorkRecord** | Defines a request to cancel a Work Record. |
| **SyncWorkRecord** | Defines a received message that may contain a new, changed, or deleted Work Record. |

## WorkRecordEntryType Group

WorkRecordEntryType is a group definition that is included in most of the other WorkRecord elements.

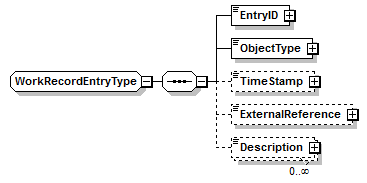


Figure - Work Record Entry Type - Group Definition

|  |  |
| --- | --- |
| **Element/Type** | **Description** |
| ***Entry ID***  IdentifierType | A unique identification of the WorkRecord individual element. This element is mandatory. |
| ***ObjectType***  WorkRecordObjectTypeType | Identifies the type of object an entry is based upon. This element is mandatory.  This may be either a standard type or an application specific extended type. Standard enumerations correspond to the WorkRecord element types and are:   * **Work Record** * **Change** * **Comment** * **Data Set** * **Event** * **Operations Definition** * **Operations Performance** * **Operations Schedule** * **Personnel Identification Manifest** * **Resource Qualifications** * **Resource Definition Manifest** * **Sample** * **Sample Test** * **Sample Test Result** * **Work Alert** * **Work Calendar** * **Work Directive** * **Work Master** * **Work Performance** * **Work Schedule** * **Other**   If “Other” then the type is an application specific extension and the value is defined in the attribute “OtherValue”. |
| ***TimeStamp***  DateTimeType | The optional time stamp associated with the entry. |
| ***ExternalReference***  IdentifierType | Contains an optional reference to data which is stored external to the WorkRecord. |
| ***Description***  DescriptionType | Additional optional additional information about the WorkRecord. |

## Work Record Element

A top level element is a WorkRecord element or type WorkRecordType.

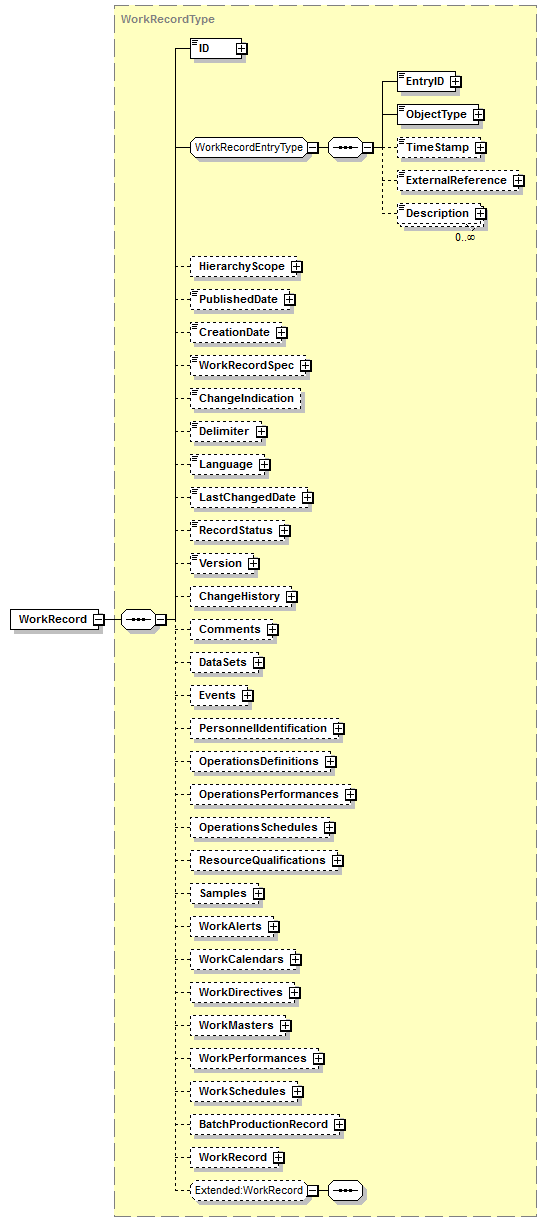


Figure - Work Record Type

| **Element/Type** | **Description** |
| --- | --- |
| ***ID***  IdentifierType | A unique identification of the WorkRecord. This element is mandatory. |
| ***ObjectType***  RecordObjectTypeType | Identifies the type of object an entry is based upon. |
| ***TimeStamp***  DateTimeType | The time stamp associated with the entry. |
| ***ExternalReference***  IdentifierType | Contains a reference to data which is stored external to the WorkRecord. |
| ***Description***  DescriptionType | Additional information about the WorkRecord. |
| ***HierarchyScope***  IdentifierType | The equipment hierarchy scope of the data associated with WorkRecord.  This information represents the physical structure of the WorkRecord to identify its context within the plant physical hierarchy |
| ***PublishedDate***  DateTimeType | The date the WorkRecord was published. |
| ***CreationDate***  DateTimeType | The date the WorkRecord was created. |
| ***WorkRecordSpec***  IdentifierType | An optional identification of the Work Record Specification that was used to generate the WorkRecord. The format for this specification is not defined. |
| ***ChangeIndicator***  xsd:string | An optional indication enabling detection that the batch production record has not been altered.   * Example 1: A string generated by an MD5 algorithm used as a hashing algorithm. * Example 2: A string representing a digital key of the entire batch production record. * Example 3: A string representing a checksum of the entire batch production record. |
| ***Delimiter***  TextType | Delimiter character used to separate equipment elements in the EquipmentID and PhysicalAssetID elements. Example: “\” |
| ***EquipmentID***  IdentifierType | A definition of the equipment associated with the WorkRecord. |
| ***Language***  CodeType | The overall language used in the WorkRecord. Note that and DescriptionType or TextType have an optional Language attribute.  Language codes should be specified using the ISO 639: 1988 specification. |
| ***LastChangedDate***  DateTimeType | The date the WorkRecord was last changed. |
| ***RecordStatus***  CodeType | Specifies the current status of the WorkRecord, reflecting the current position in the WorkRecords life cycle. There are no standard codes defined.  Example: In Process, In Review, Approved. |
| ***Version***  IdentifierType | The current version of the WorkRecord. |
| ***ChangeHistory***  ChangeHistoryType | Container type for Change elements.    Each Change element contains:  ***RecordReference*** (IdentifierType): The EntryID of a specific element in the WorkRecord.  ***PreChangeData*** (ValueType): A definition of the value of a data element prior to change of the data element.  ***Reason*** (TextType): A text comment with the reason for the change.  WorkRecord_diagrams/WorkRecord_p77.png |
| ***Comments***  CommentsType | Container type for Comment elements.    Each Comment element contains:  ***RecordReference*** (IdentifierType): The EntryID of a specific element in the WorkRecord.  ***CommentText*** (TextType): The text comment.  ***PersonID*** (NameType): Defines the person associated with the comment.  WorkRecord_diagrams/WorkRecord_p83.png |
| ***DataSets***  DataSetsType | Container type for DataSet elements.    Each DataSet element contains:  ***TrendSystemReference*** (IdentifierType): Specifies the location of the data set if is stored on an external system.  ***StartTime*** (DateTimeType): Date and time of the start of data in the data set.  ***EndTime*** (DateTimeType): Date and time of the end of data in the data set.  ***TimeSpecification*** (TimeSpecificationType): Defines the attributes of the time specification of the data.  ***TagSpecification*** (TagSpecificationType):  ***DelimitedDataBlock*** (DelimitedDataBlockType):  ***OrderedData*** (OrderedDataType):  WorkRecord_diagrams/WorkRecord_p89.png |
| ***Events***  EventsTypes | Container type for Event elements.    The event contains an EventType, EventSubtype, and value..  Other elements include:  ***EquipmentID*** (IdentifierType): Defines the equipment associated with the event.  ***Value*** (ValueType):A value associated with the event  ***PreviousValue*** (ValueType):A previous value associated with the event  ***MessageText*** (TextType): Defines text associated with the event.  ***PersonID*** (NameType): Defines an identification of the person associated with the event.  ***ComputerID*** (IdentifierType): Defines the computer or automation system associated with the event.  ***PhysicalAssetID*** (IdentifierType): Defines the physical asset associated with the event.  ***JobStepReference*** (IdentifierType): Defines a reference to a procedural element associated with the event, such as a phase or an operation.  ***Category*** (IdentifierType): Defines the use category of the event. No standard identifiers are defined. Example; Informational, Critical.  ***AlarmData*** (AlarmDataType): Defines additional alarm data if the event is an alarm type.  ***AssociatedEventID*** (IdentifierType): Defines the EntryID of any associated event.  ***UserAttribute*** (UserAttributeType):Defines additional user defined attributes associated with the event.  WorkRecord_diagrams/WorkRecord_p107.png |
| ***PersonnelIdentification***  PersonnelIdentificationType | Container type for PersonnelIdentificationManifest elements.    A PersonnelIdentification Manifest contains  **RecordReference** (IdentifierType): Defines the EntryID of a record in the WorkRecord that the personnel identification manifest is associated with, for example a Change element.  **Name** (NameType): Defines the name of the person.  **ChangeIndication** (xsd:string): An indication enabling detection that the PersonnelIdentificationManifest element has not been altered. Example: string generated by an MD5 hash algorithm of the name, record reference, and reason.  **Reason** (TextType): Defines the reason for the signature.  WorkRecord_diagrams/WorkRecord_p122.png |
| ***OperationsDefinitions***  OperationsDefinitionsType | Container type for OperationsDefinition elements.  See **B2MML-V0600-OperationsDefinition** for a specification of the OperationsDefinition element type.    WorkRecord_diagrams/WorkRecord_p230.png |
| ***OperationsPerformances***  OperationsPerformancesType | Container type for OperationsPerformance elements.  See **B2MML-OperationsPerformance** for a specification of the OperationsPerformance element type.    WorkRecord_diagrams/WorkRecord_p234.png |
| ***OperationsSchedules***  OperationsSchedulesType | Container type for OperationsSchedule elements.  See **B2MML-OperationsSchedule** for a specification of the OperationsSchedule element type.    WorkRecord_diagrams/WorkRecord_p238.png |
| ***ResourceQualifications***  ResourceQualificationsType | Container type for ResourceQualificationManifest elements.    A rESOURCEqUALIFICAITONmANIFEST CONTAINS:  **RecordReference** (IdentifierType): Defines the EntryID of a record in the WorkRecord that the resource qualification manifest is associated with  **ResourceID** (IdentifierType): Defines the resource associated with the WorkRecord element.  **QualificationStatus** (CodeType): Defines the criteria of the resource that was recorded. No standard codes are defined. Examples: Sterilized, Cleaned, Services.  **EffectiveTimeStamp** (DateTimeType): Defines the date and time the qualification became effective  **ExpirationTimeStamp** (DateTimeType): Defines the date and time the qualification expired.  WorkRecord_diagrams/WorkRecord_p164.png |
| ***Samples***  SamplesTypes | Container type for Sample elements.    A Sample defines:  Defines that a material was sampled and/or tested and results of the test  **SampleSourceID** (IdentifierType): Defines the lot, batch, or sample that the sample material was pulled from.  **SampleSize** (QuantityValueType): Defines the amount of material taken for the sample.  **SampleType** (CodeType): Defines the type of sample pulled. There are no standard codes defined. Examples: Receiving Sample, Online Sample, Lab Sample,  **SamplePullReason** (TextType): Defines the reason the sample was pulled.  **SampleExpiration** (DateTimeType): Defines the date the sample expires.  **EquipmentID** (IdentifierType): Defines a reference to an equipment element that is associated with the sample. Example: The Unit the sample was pulled from.  **PhysicalAssetID** (IdentifierType): Defines a reference to a physical asset element that is associated with the sample.  **SOPReference** (IdentifierType): Defines the standard operating procedure (SOP) that is associated with the sample pull.  **SampleTest** (SampleTestType): Defines the tests on the sample.  WorkRecord_diagrams/WorkRecord_p166.png |
| ***BatchProductionRecord***  BatchProductionRecordType | Contains a BatchProductionRecord. |
| ***WorkAlerts***  WorkAlertsType | Container type for Work Alerts.  WorkRecord_diagrams/WorkRecord_p147.png  WorkRecord_diagrams/WorkRecord_p214.png |
| ***WorkCalendars***  WorkCalendarsType | Container type for Work Calendars.  WorkRecord_diagrams/WorkRecord_p148.png  WorkRecord_diagrams/WorkRecord_p218.png |
|  |  |
| ***WorkMaster***  WorkMasterRecordType | Container type for Work Master elements |
| ***WorkDirective***  WorkDirectiveRecordType | Container type for Work Directive elements |
| ***WorkSchedule***  WorkScheduleRecordType | Container type for Work Schedule elements |
| ***WorkPerformance***  WorkPerformanceRecordType | Container type for Work Performance elements |

## Data Elements

| **Element/*Type*** | **Description** |
| --- | --- |
| ***AlarmData***  WorkAlarmDataType | Defines additional alarm data for an event that is an alarm event.  ***AlarmEvent*** (CodeType): Defines the type of alarm event. Standard identifiers are not defined. Examples: Detected, Acknowledged, and Cleared.  ***AlarmType*** (CodeType): Defines the type of alarm. Standard identifiers are not defined. Examples: High, Deviation, Rate of Change.  ***AlarmLimit*** (ValueType): Defines the value measurement that caused the alarm event.  ***Priority*** (IdentifierType):Defines the importance of the event. Standard identifiers are not defined. Examples: High, Low, 10, 9,8, 0 …  WorkRecord_diagrams/WorkRecord_p63.png |
| ***DataValue***  DataValueType | Defines the value for a data element in a data set.  ***TagIndex*** (NumericType):Defines a numeric index value the Tag element. NOTE: This is required because the order of repeating elements is not guaranteed in XML.  ***Value*** (xsd:string): Defines the value of the data for a tag for a data set.  ***Quality*** (IdentifierType): Defines the quality of the value. This indicates the lack or presence of problems associated with the collection of the data. There are no standard identifiers defined. Example: Good, Uncertain, Out of Date. |
| ***DelimitedDataBlock***  DelimitedDataBlockType | Defines a data set in a single delimited string element. The string contains a set of substrings, delimited by the OrderDelimiter, and tag elements in the substring delimited by the TagDelimiter. The selection of the TagDelimiter and OrderDelimiter will be dependent on the data collected. The delimiter characters cannot be part of the data.  Note: This data representation is not directly defined in ANSI/ISA 88.04, but is provided to have a more compact form of data representation.  ***TagDelimiter*** (): Defines the character used to delimit the tag elements within an Order substring. Example: “\”.  ***OrderDelimiter*** (): Defines the character used to delimit the order substrings. Example “|”.  ***DelimitedData*** (): Defines the delimited string with tag elements. Example:  “09:10:21\23.4\57\76.2|09:10:25\23.3\57\76.1|09:10:27\23.2\57\76.0””  Empty tag data values are indicated by no spaces between TagDelimiters. |
| ***EventSubType***  EventSubTypeType | Defines an event subtype. The associated meaning of the event type, subtype, value, and previous value are specified in ANSI/ISA 88.04 and IEC 61512-4.  This may be either a standard type or an application specific extended type. Standard enumerations correspond to the WorkRecord element types and are:   * **Allocation** * **Application** * **Consume** * **Deallocation** * **Equipment** * **Message** * **Mode Change** * **Mode Command** * **Modification** * **Movement** * **Parameter Data** * **Process** * **Process Data** * **Produce** * **Prompt** * **Prompt Response** * **Property Value Change** * **Reconciliation** * **Security** * **State Change** * **State Command** * **Status Change** * **System** * **Target End Time** * **Target Start Time** * **Other**   If “Other” then the type is an application specific extension and the value is defined in the attribute “OtherValue”. |
| ***EventType***  EventTypeType | Defines an event type. The associated meaning of the event type, subtype, value, and previous value are specified in ANSI/ISA 88.04 and IEC 61512-4.  This may be either a standard type or an application specific extended type. Standard enumerations correspond to the WorkRecord element types and are:   * **Alarm** * **Equipment** * **General** * **Material** * **Message** * **Operator** * **Procedural Execution** * **Work Directive** * **Other**   If “Other” then the type is an application specific extension and the value is defined in the attribute “OtherValue”. |
| ***OrderedData***  OrderedDataType | Defines a set of ordered data values.  ***OrderIndex*** (NumericType): Defines a numeric index value the OrderedData element if there is not TimeValue defined. NOTE: This is required because the order of repeating elements is not guaranteed in XML.  ***TimeValue*** (DateTimeType): Defines the date and time of the OrderedData element if it is time base.  ***DataValue*** (DataValueType): Defines the value of the data,  WorkRecord_diagrams/WorkRecord_p97.png |
| ***SampleTest***  SampleTestType | Defines information on a test of a sample.  **TestCode** (CodeType): defines the test procedure used. Example: CHR3390  **TestName** (IdentifierType): Defines the name associated with the test performed. Example: Bio Burden  **SampleTestResult** (SampleTestResultType): |
| ***SampleTestResult***  SampleTestResultType | Defines the result from a sample test.  **TestDisposition** (IdentifierType):Defines an indication if the test results were acceptable. No standard identifiers defined. Example: Pass, Fail  **EquipmentID** (IdentifierType): Defines the equipment used to perform the tests.  **PhysicalAssetID** (IdentifierType): Defines the physical asset used to perform the tests.  **AnalysisUsed** (CodeType): Defines the statistical sampling analysis used to determine the result. No standard CodeType is defined. Example: Average, Minimum, Maximum  **Expiration** (DateTimeType): Defines the date and time that the sample test results expire.  **Results** (ValueType): Defines the actual value or values returned from the test.  **ExpectedResults** (ValueType): Defines the expected value or values returned from the test. |
| ***TagSpecification***  TagSpecificationType | Defines a unique identification of a data source. It contains the information to identify a single data source (called a tag) and any data required to interpret the data values, such as data compression information required to use the data.  ***TagIndex*** (NumericType): An internal number that is used to identify the specific data value in a dataset. (Because the order of repeating elements in XML is not guaranteed, this number is used to tie the data value to the tag specification,)  ***DataType*** (DataTypeType): From the common elements, defines the data type of the data. (Example: long, unsignedInt, float,…)  ***UnitOfmeasure*** (UnitOfMeasureType): From the Common elements, defines the unit of measure as a CodeType.  ***DataSource*** (IdentifierType): Defines the data source for the data element, typically the tag name or equivalent.  ***Alias*** (IdentifierType): Defines an alternate ID of the data source of the data element.  ***Description*** (DescriptionType): Defines additional information about the data source or the alias identification.  ***EquipmentID*** (IdentifierType): Defines a reference to equipment, This information represents the physical structure above the referenced entity to identify its context within the plant physical hierarchy.  ***PhysicalAssetID*** (IdentifierType): Defines a reference to a physical asset.  ***ProceduralElementReference*** (IdentifierType): Reference to the procedural element, such as procedure, unit procedure, operation, recipe phase, equipment phase, or equipment step associated with the data source ID for the time period of the trend segment.  ***Deadband*** (ValueType): A deadband value used for collecting and storing data values. If different high and low deadbands are specified, then they should be separate values with keys of HIGH and LOW.  ***SignificantDigits*** (IdentifierType): Defines the significant digits used for collecting and storing data. This is kept as an identifier type to allow specification or formats for different data types,, such as F2.4 or F7.1 for floating point information and I5 or I8 for integer information.  ***DataCompression*** (IdentifierType): An identification of the data compression algorithm used when the data was collected. There are no standard identifiers defined. Example; Boxcar Backslope, Change Delta, None.  ***SamplingType*** (IdentifierType): The type of sample stored. There are no standard identifiers define. Example: Actual/Raw, Interpolated, Best Fit. |
| ***TimeSpecification***  TimeSpecificationType | Defines the attributes of the time specification of the data.  ***Relative*** (IndicatorType): If TRUE then the date/time is relative time, if FALSE then the date/time is absolute time.  ***OffsetTime*** (DateTimeType): If the time is relative, then the offset defines the absolute time that may be added to the relative times to determine the absolute times.  . |
| ***UserAttribute***  UserAttributeType | Defines user defined attributes and values for events. |

# Diagram Convention

The schema diagrams using the following convention to illustrate the structure of the schema elements, the type of the elements and attributes, and the rules for optional elements and repetition.



About MESA: MESA promotes the exchange of best practices, strategies and innovation in managing manufacturing operations and in achieving operations excellence. MESA’s industry events, symposiums, and publications help manufacturers achieve manufacturing leadership by deploying practical solutions that combine information, business, manufacturing and supply chain processes and technologies. Visit us online at <http://www.mesa.org>.

About the XML Committee: The XML Committee was formed within MESA to provide a forum for the development of the B2MML and BatchML specifications.