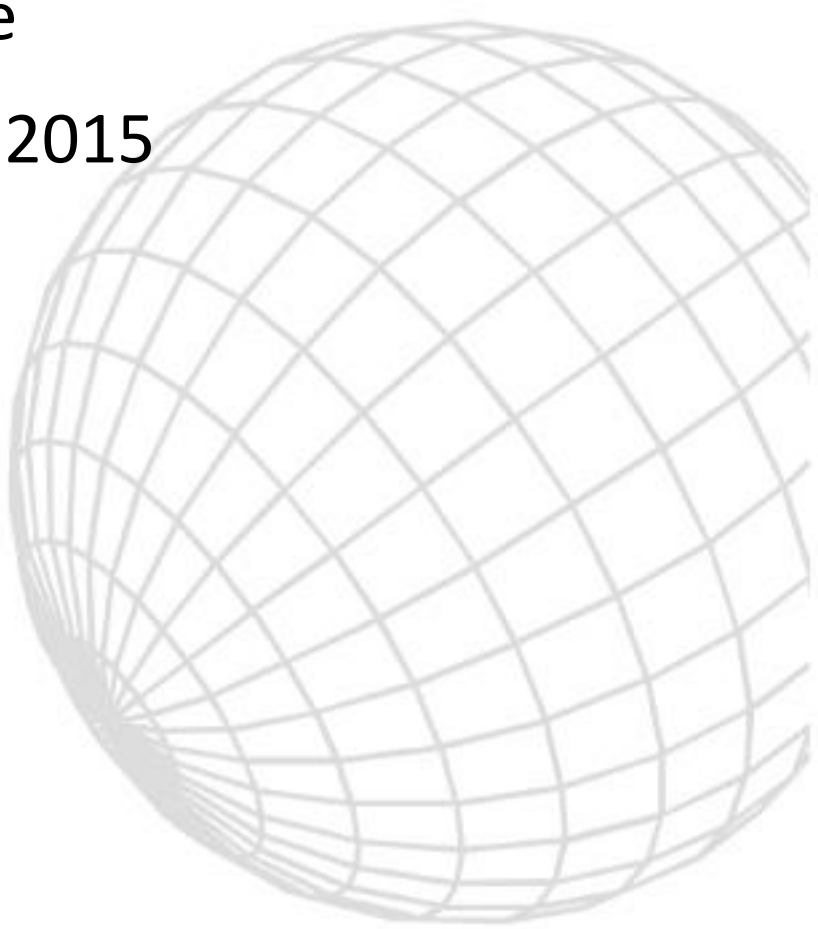
MESA INTERNATIONAL
Driving Operations Excellence

Key Performance Indicator Markup Language

Version 01 - May 2015



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.

Table of Contents

1 CHANGE HISTORY

Change	Date	Person	Description
V01	1 May 2015	D. Brandl	Initial version

2 REFERENCES

[1] ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod), Enterprise-Control System Integration − Part 1: Models and Terminology
[2] ANSI/ISA-95.00.02-2010 (IEC 62264-2 Mod), Enterprise-Control System Integration − Part 2: Object Model Attributes
[3] ANSI/ISA-95.00.05-2013, Enterprise-Control System Integration − Part 5: Business-to-Manufacturing Transactions
[4] ANSI X12 EDI Allowable Units of Measure and Codes
[5] IETF RFC 2045 – Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
[6] IETF RFC 2046 – Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
[7] IETF RFC 2047 – Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text
[8] ISO 639-1:2002 – Codes for the representation of names of languages – Part 1: Alpha-2 code
[9] ISO 639-2:1998 – Codes for the representation of names of languages – Part 2: Alpha-3 code descriptions of KPIs
[10] ISO 3166-1:2013 – Codes for the representation of names of countries and their subdivisions - Part 1: Country codes
[11] ISO 3166-2:2013 – Codes for the representation of names of countries and their subdivisions - Part 2: Country subdivision code
[12] ISO 3166-3:2013 – Codes for the representation of names of countries and their subdivisions - Part 3: Code for formerly used names of countries
[13] ISO 4217:2008 – Codes for the representation of currencies and funds
[14] ISO 8601:2004 – Data elements and interchange formats - Information interchange - Representation of dates and times
[15] ISO 22400-1 Automation systems and integration — Key performance indicators (KPIs) for manufacturing operations management — Part 1: Overview, concepts and terminology
[16] ISO 22400-2 Automation systems and integration — Key performance indicators (KPIs) for manufacturing operations management — Part 2: Definitions and descriptions
[17] ISO 80000-1:2011 – Quantities and units – Part 1: General Corrigendum 1
[18] OAGiS – The Open Application Group Interface Specification – www.oagi.org
[19] Professional XML Schemas, 2001, published by WROX (ISBN 1-861005-47-4
[20] UN/CEFACT - United Nations framework of the Economic and Social Council, Core Components Library
[21] UN/ECE Recommendation 20 – Code for units of measure used in international trade
[22] UN/ECE Recommendation 21 – Code for types of cargo packages and packaging materials with complementary codes for package names
[23] UN/EDIFACT – United Nations rules for Electronic Data Interchange for Administration, Commerce and Transport, Data element 3055 Code List
[24] MESA B2MML – Business to Manufacturing Markup Language – www.mesa.org

3 SCOPE

This document defines and presents the Key Performance Indicator Markup Language, KPI-ML. KPI-ML is based on XML and KPI-MLs schemas are intended to hold information related to Key Performance Indicators (KPIs). The schemas allow for a standard representation of generic KPI definitions, specific instances of use of KPIs (such as related to specific equipment), and KPI values for specific KPI instances.

The information in this document is based on the data models and attributes defined in the ISO 22400 standards and the ANSI/ISA 95 standards.

Contact ISO (International Standardization Organization) and ISA (The International Society of Automation) for copies of the standards. Additional information on these standards is also available at www.iso.org and www.isa.org.

4 SCHEMA SCOPE

The KPI-ML schemas are based on the standards ISO 22400-1 Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management - Part 1 Overview, concepts and terminology [14] and ANSI/ISA-95.00.05-2006 Enterprise-Control System Integration Part 5: Business to Manufacturing Transactions [3].

An example of KPI-ML using the KPI's defined in the ISO 22400-2 [15] KPI definitions is also included as part of the KPI-ML deliverables. See Section 9 for additional information,

This information is based on the data models and attributes defined in the ISO 22400 KPI Standard and ANSI/ISA 95 Enterprise/Control System Integration standard. Contact ISO and ISA (The International Society of Automation) for copies of the standard. Additional information on these standards is available at www.iso.org and www.isa.org.

4.1 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 = "Scope" type = " ScopeType"/>

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

The method is the "Venetian Blind Model" [19] for schema definitions. This model makes all of the type names global and usable in user derived works, without a loss of context or additional information required to identify the element as of being of the same type as related KPI-ML elements.

4.2 User Element Extensibility

In order to make the schemas more useful, selected elements 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 section 10 for a complete explanation of user extensibility.

4.3 Use of IDs and Names in Schema Definitions

The use of **IDs** in the schema definition is based on the definition of **ID** attributes in the ISO 22400-1 [14] standard. The ID usually defines a specific instance of an element in an exchanged XML file. When defining KPIDefinitions and KPIInstances, the ID should be a persistent value that can be used to relate KPIInstances to KPIDefinitions and KPIValues to KPIInstances.

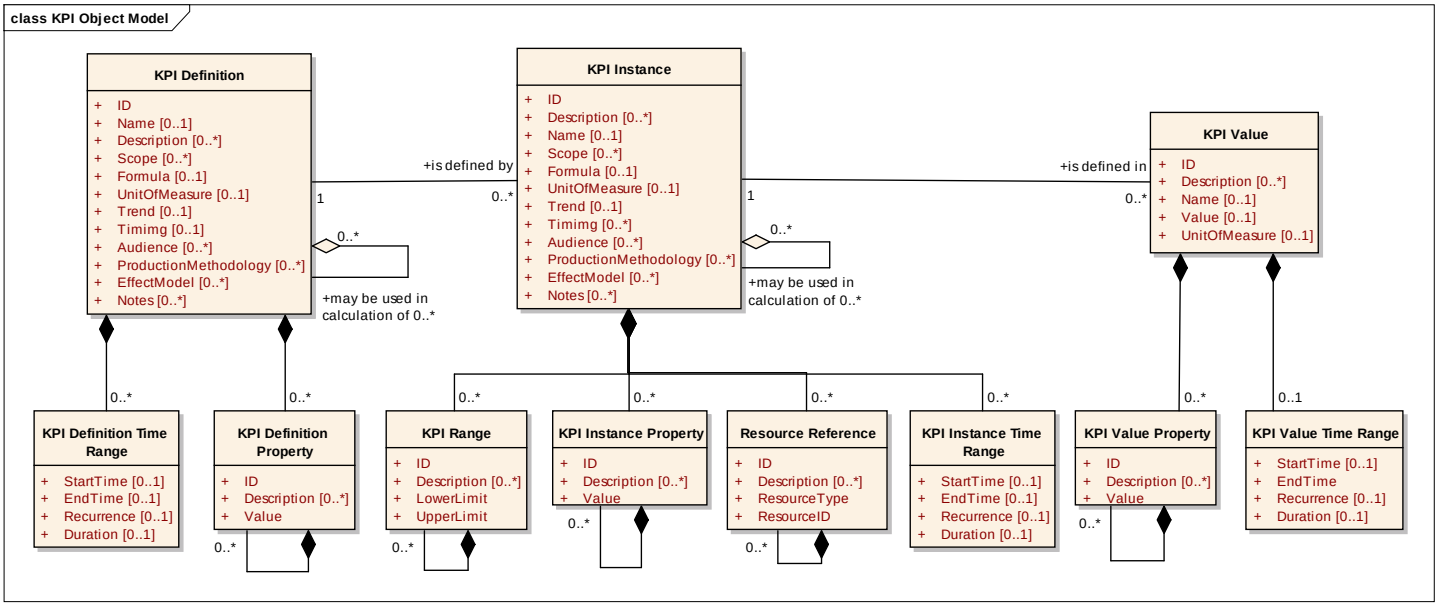
The use of **names** in the schema definition is based on the definition of **name** attributes in the ISO 22400-1 standard [14]. Names are meant to be a human readable form of the ID. The **name** should not be considered unique across all exchanges.

4.4 KPI Object Model

The KPI object model is derived from the object model specified in ISO 22400-1 [14] with the following changes:

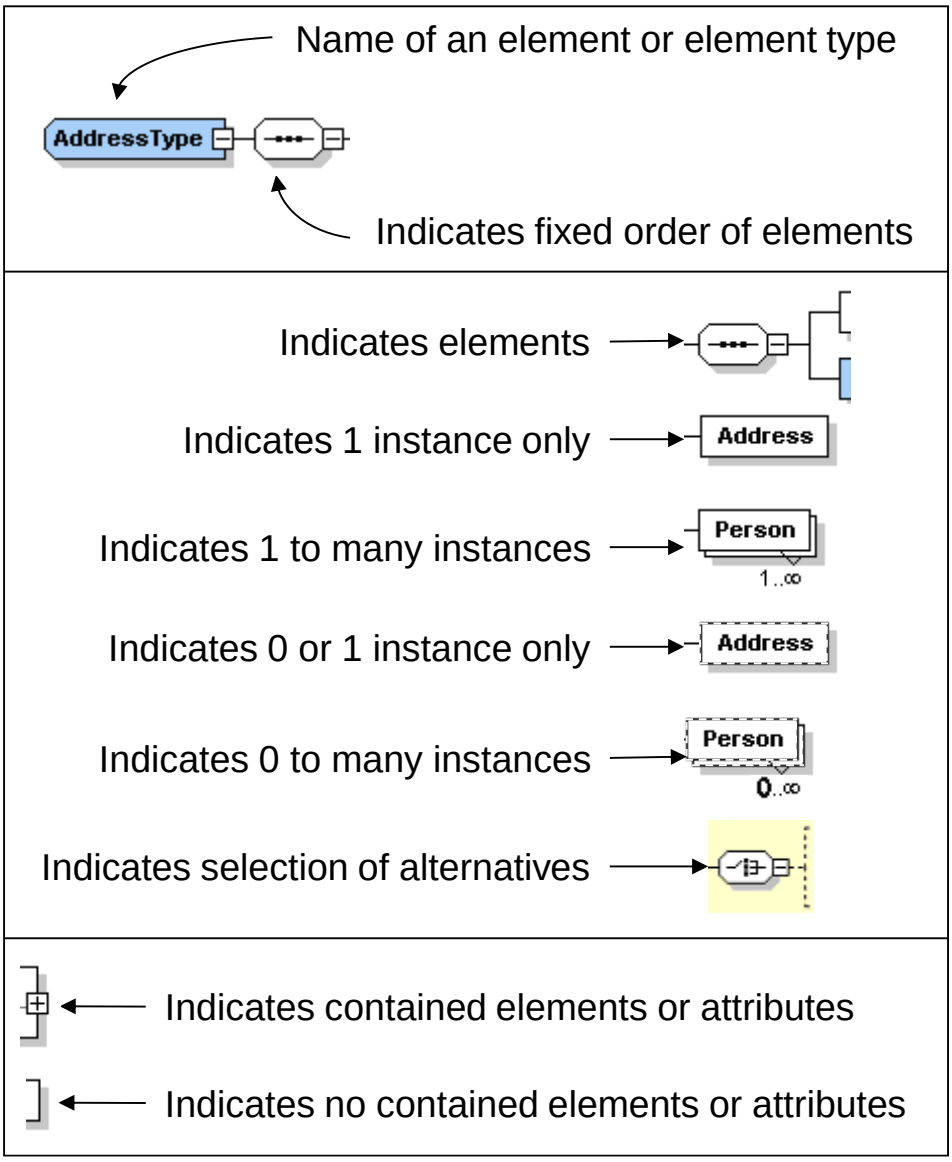
1. *KPI Values* may have properties. KPI Value Properties are optional and are designed to be used to exchange information about the measurements or metrics used to calculate the KPI value.
2. *KPI Definition Property*, *KPI Instance Property*, and *KPI Value Property* objects may be recursive. This is consistent with the property definitions in the ISA 95.02, IEC 62264-2 [2], and with the MESA B2MML schemas.
3. A time range was added to specify detailed time ranges and intervals within the ranges. The time range contains a start time, an end time, a recurrence interval, and a duration. Simple timing contains only a start time and end time. More complex timing can use the recurrence and duration to specify specific intervals within the start and end time that a KPI is collected. All times are defined using the ISO 8601 format. A KPI Definition and KPI Instance may specify 0 or more time intervals, while a KPI Value may specify only one range or interval.
4. The hierarchy of KPIs defined in ISO 22400-1 and ISO 22400-2, where one or more KPI values are used to calculate a higher level KPI, are represented as the “may be used in calculation” associations, and represented in KPI-ML with the UsesKPIDefinitionID and UsesKPIInstanceID elements.

   For example: OEE can be one element of a strategic KPI at the enterprise level, in which case, a representation of the hierarchy of KPIs can be represented in KPI-ML.



4.5 Diagram Convention

The schema diagrams use the convention in the figure below to illustrate the structure of the schema elements, the type of the elements and attributes, and the rules for optional elements and repetition.



5 UN/CEFACT CORE COMPONENT TYPES

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 [20] and ISO TC 154.

NOTE: The core components contain optional attributes that may be used to specify the context and source of the associated element value. All attributes are optional in KPI-ML.

The core components use several international standards for the representation of semantic and standardized information:

Name	Standard
Country Code	ISO 3166.1 [10]
Region Code	ISO 3166.2 [11]
Language Code	ISO 639: 1988 [8],[9]
Currency Code	ISO 4217 [13]
Date and Time Representation	ISO 8601 [14]
Unit Of Measure Code	UN/ECE Recommendation 20 [21]
Unit of Transport or Packaging Code	UN/ECE Recommendation 21 [22]

5.1 BinaryObjectType

BinaryObjectType is used to define data types representing graphics, pictures, sound, video, or other forms of data that can be represented as a finite length sequence of binary octets. It is derived from base64Binary.

Optional Attribute	Base XML Type	Description
Format	string	The format of the binary content. No identifiers for standard formats are defined.
mimeCode	normalizedString	The mime type of the binary object. See IETF RFC 2045, 2046, and 2047. [5],[6],[7]
encodingCode	normalizedString	Specifies the decoding algorithm of the binary object. See IETF RFC 2045, 2046, and 2047. [5],[6],[7]
characterSetCode	normalizedString	The character set of the binary object if the mime type is text. See IETF RFC 2045, 2046, and 2047. [5],[6],[7]

Optional Attribute	Base XML Type	Description
Uri	anyURI	The Uniform Resource Identifier that identifies where the binary object is located.
Filename	string	The filename of the binary object. See IETF RFC 2045, 2046, and 2047. [5],[6],[7]

5.2 CodeType

CodeType is used to define a character string that is used to represent an entry from a fixed set of enumerations. It is derived from the type **normalizedString**.

All of the KPI-ML enumerations are derived from **CodeType**. Also, KPI-ML elements that are not identifications of objects or other elements are derived from **CodeType**.

Optional Attribute	Base XML Type	Description
listID	normalizedString	An Identifier specifying the identification of a code list that this is registered with at an agency. For example: UN/EDIFACT data element 3055 code list [23]
listAgencyID	normalizedString	An Identifier specifying the agency that maintains one or more lists of codes. For example: INDICOD, NABCA.
listAgencyName	string	Text that contains the name of the agency that maintains the list of codes. For example: Istituto Nazionale per la Diffusione della Codifica dei Prodotti, National Alcohol Beverage Control Association
listName	string	Text that contains the name of a code list that this is registered with at an agency. For example: CSC (for the NABCA)
listVersionID	normalizedString	An Identifier specifying the version of the code list.
Name	string	Text equivalent of the code content component. For example “Control State Code”
languageID	language	An Identifier specifying the language used in the code name. For example: “EN”
listURI	anyURI	The Uniform Resource Identifier (URI) that identifies where the code list is located. For example: “www.nabca.org”
listSchemaURI	anyURI	The Uniform Resource Identifier (URI) that identifies where the code list scheme is located. For example “www.nabca.org/StatisticalData/AccountLevel.aspx”

5.3 DateTimeType

DateTimeType is used to define a particular point in time together with the relevant supplementary information to identify the timezone information. It is derived from the type **dateTime**. In KPI-ML this is a specific instance on time using the ISO 8601 CE (Common Era) [14] calendar extended format and abbreviated versions. For example:

yyyy-mm-ddThh:mm:ssZ for UTC as “2002-09-22T13:15:23Z”

Optional Attribute	Base XML Type	Description
Format	string	Not needed in KPI-ML, but maintained for compatibility with OAGiS. [18] A string specifying the format of the date time content, however the format of the format attribute is not defined in UN/CEFACT [20] specification.

5.4 IdentifierType

IdentifierType is used to define a character string to identify and distinguish uniquely, one instance of an object in an identification scheme from all other objects in the same scheme. It is derived from the type **normalizedString**.

All of the KPI-ML ID types are derived from **IdentifierType**.

Optional Attribute	Base XML Type	Description
schemaID	normalizedString	An Identifier specifying the identification of the identification schema.
schemaName	string	Text that contains the name of the identification scheme.
schemaAgencyID	normalizedString	An Identifier specifying the identification of the agency that maintains the schema.
schemaAgencyName	string	Text containing the identification of the agency that maintains the schema.
schemaVersionID	normalizedString	The version (as an Identifier) of the schema.
schemaDataURI	anyURI	The Uniform Resource Identifier (URI) that identifies where schema data is located.
schemaURI	anyURI	The Uniform Resource Identifier (URI) that identifies where schema is located.

5.5 MeasureType

MeasureType is used to define a numeric value determined by measuring an object along with the specified unit of measure. It is derived from type **decimal**.

Optional Attribute	Base XML Type	Description
unitCode	normalizedString	The type of unit of measure. See UN/ECE Recommendation 20 [21] and ANSI X12 355 [4]
unitCodeListVersionID	normalizedString	The version of the unit of measure code list.

5.6 TextType

TextType is used to define a character string (i.e. a finite set of characters) generally in the form of words of a language. It is derived from the type **string**.

Optional Attribute	Base XML Type	Description
languageID	language	An Identifier specifying the language used in the content component.
languageLocaleID	normalizedString	An Identifier specifying the locale of the language

5.7 String Type Usage

The support for UN/CEFACT [20] core components and compatibility with OAGiS [18] has required the use of three basic string types, each with separate purposes:

1. CodeType is required to be compatible with the core components
2. xsd:normalizedString is required to be compatible with OAGiS [18] transaction processing
3. xsd:string is required to hold special characters (tab, LF, CR)

CodeType

- CodeType is used anyplace there is an enumeration.
- This follows the UN/CEFACT [20] standard, it provides attributes that can be used to identify who “owns” the enumeration.
- This is derived from the xsd:normalizedString.

xsd:normalizedString

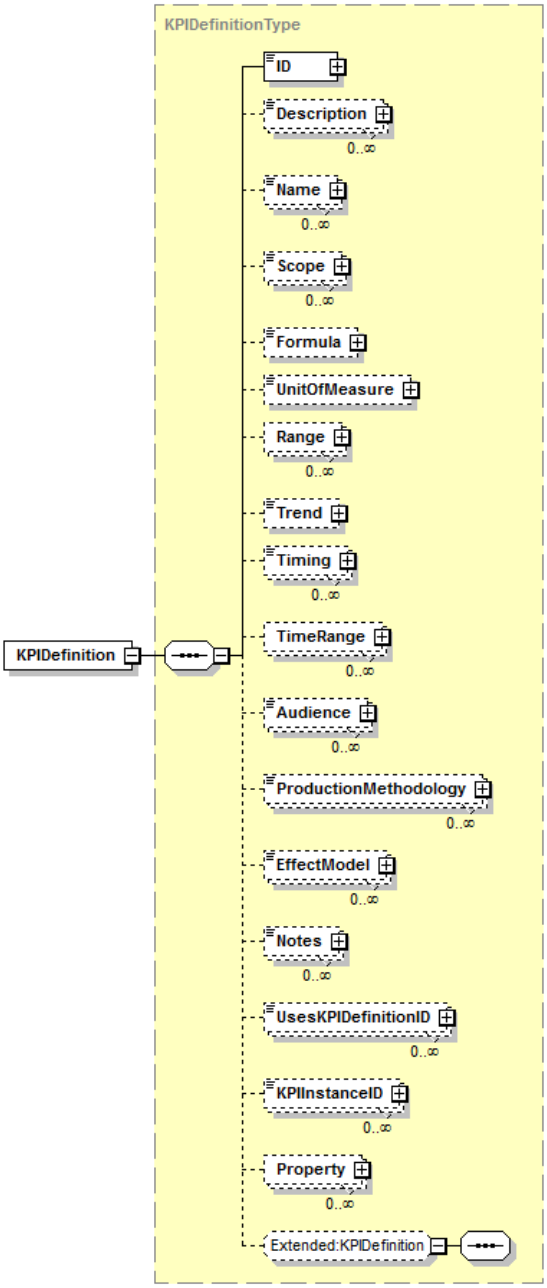
- xsd:normalizedString is a string in which line feeds, carriage returns, and tabs have been replaced by blanks. There can be multiple blanks in the string.
- This is used in KPI-ML for all of the attributes defined in the core component types. This should not be changed because it would no longer match the recommended Core Component types.
- This is also used in the transaction elements in order to match the definition in the OAGiS [18] schemas. If this were changed, then KPI-ML would no longer be compatible with the OAGiS transaction model. It would probably not be a problem to change this to xsd:string, **BUT** it could make it difficult to find compatibility issues (for example someone uses a tab instead of a space, or has a non-printing CR in a string that causes it not to match the expected string.)

xsd:string

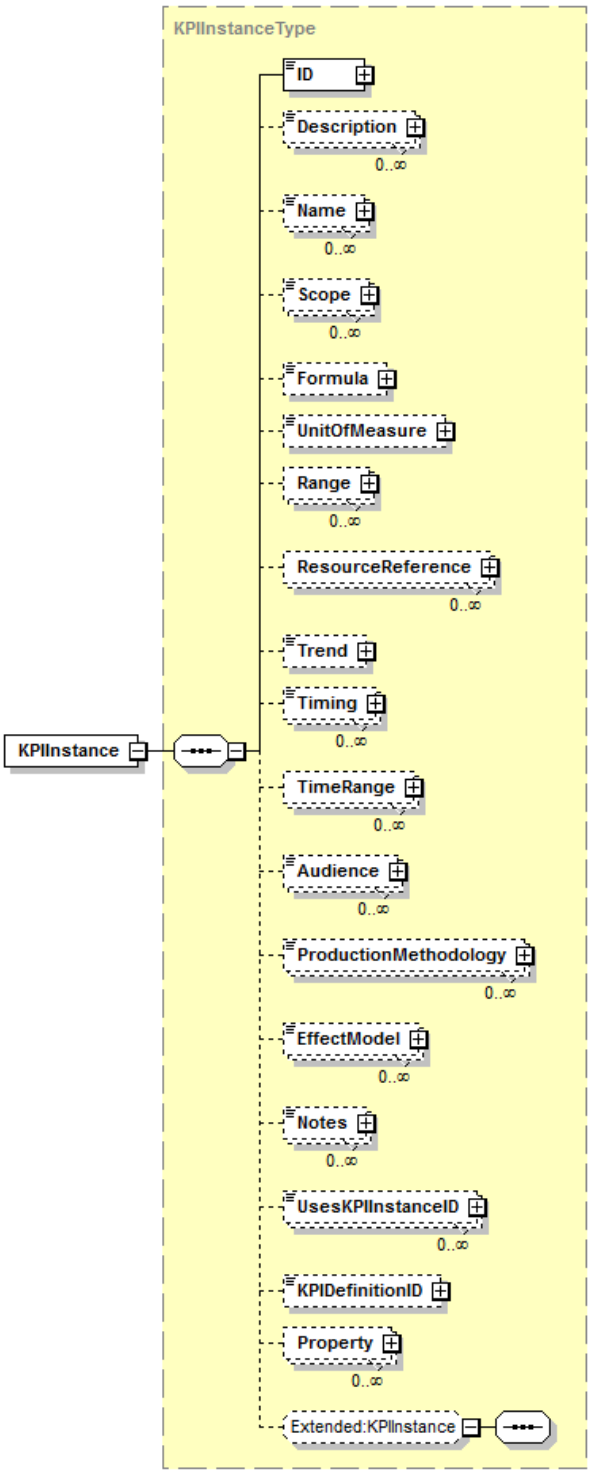
- xsd:string is a string which may contain line feeds, carriage returns, and tabs.
- This includes tag delimiters, order delimiters delimited data, and all “otherValue” attributes in enumerated lists. These should not be changed, because the tabs, LF and CR characters are important. The “otherValue” types could probably be changed to xsd:normalizedString without any major impact, because these are usually just vendor specific enumerations.

6 BASIC ELEMENT DEFINITIONS

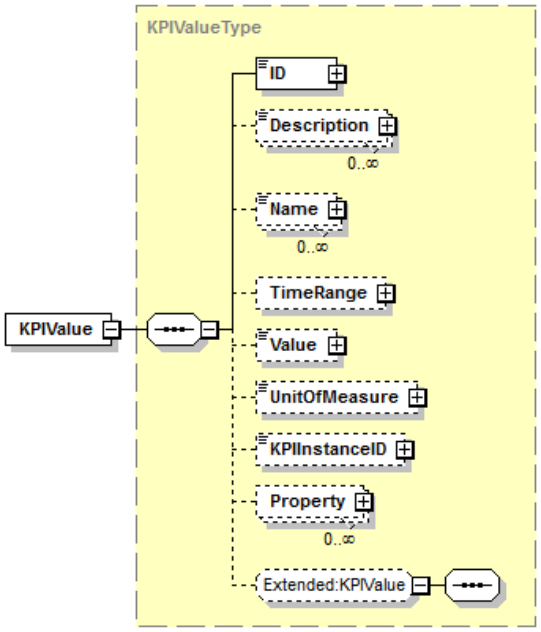
6.1 KPI Definition

Type Name	Description
KPIDefinition **KPIDefinitionType**	Contains the definition of a Key Performance Indicator. This defines the formula and general information on a KPI definition. See *ISO 22400-2 Manufacturing operations management — Key performance indicators — Part 2: Definitions and descriptions of KPIs* [15] for a list of standard KPI definitions. 

6.2 KPI Instance

Type Name	Description
KPIInstance **KPIInstanceType**	Defines an instance of use of a KPI. The instance is tied to specific resources (site, area, work center, work unit, personnel, material, organization, etc). A KPI Instance should have an equivalent specification in a KPI Definition. 

6.3 KPI Value

Type Name	Description
KPIValue **KPIValueType**	Defines a specific value for a specific time, time range, ot range and interval, for a KPI. The KPI value may contain properties. 

The Value element is optional to allow the use of a KPIValue in a GET transaction, where one or more KPIValue elements with no values, but with time ranges or KPIInstance IDs, are sent with a GET message and the response SHOW message contains the KPI's value.

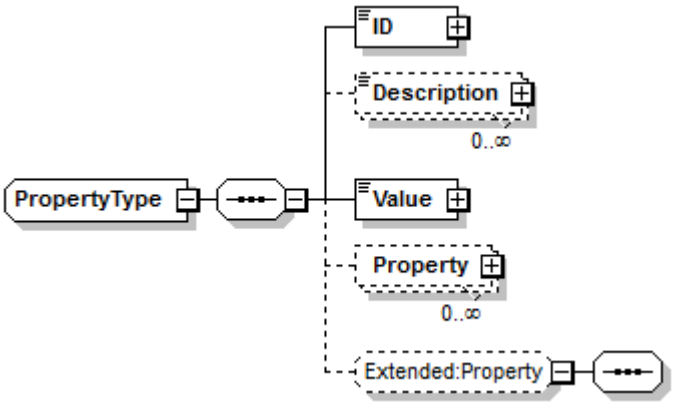
Properties for KPIValues are optional, but if used the properties should only contain the values used in calculation of the KPI. In addition a property may also be defined to contain the "Formula", duplicating the KPIInstance information.

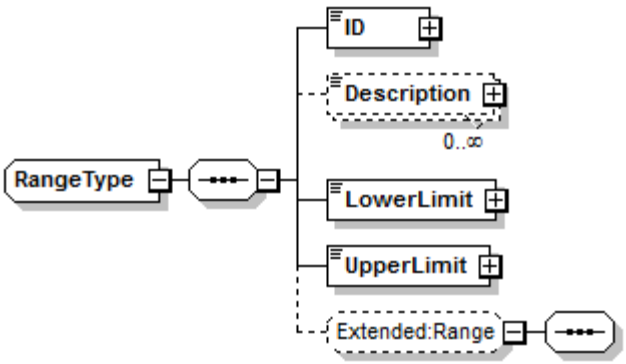
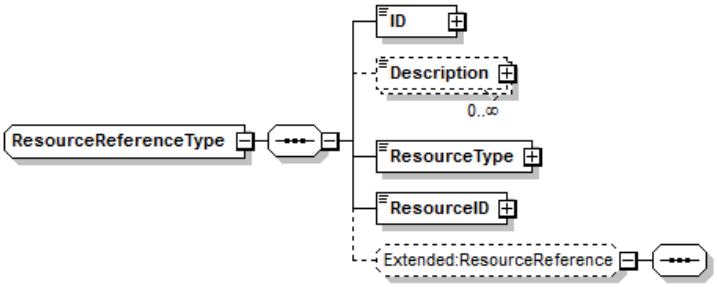
For example:

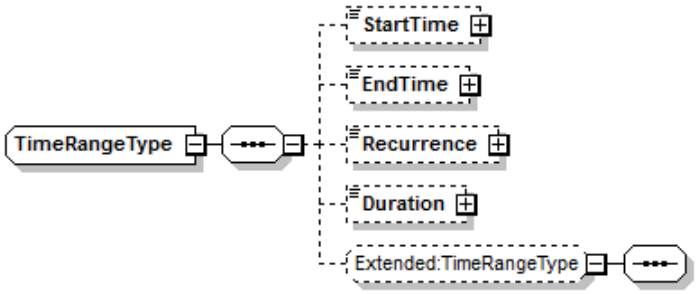
KPIValue/ID	→ 3421	
KPIValue/Name	→ WCP02WC015	
KPIValue/KPIInstanceID	→ Worker Efficiency – Plant 2, WorkCenter 15	
KPIValue/Value	→ 80	
KPIValue/UnitOfMeasure[1]	→ "%"	
KPIValue/Description	→ "the relationship between the Actual Personnel Work Time (APWT) related to production orders and the Actual Personnel Attendance Time (APAT) of the employee"	
KPIValue/StartTime	→ 2014-08-04 08:00.00	
KPIValue/EndTime	→ 2014-08-08 17:00.00	
KPIValue/Property/ID	→ Formula,	KPIValue/Property/Value → APWT/APAT
KPIValue/Property/ID	→ APWT ,	KPIValue/Property/Value → 32
KPIValue/Property/ID	→ APAT ,	KPIValue/Property/Value → 40

[1] Where possible the Unit Of Measures should be specified using The International System of Units (abbreviated SI from French: Le Système international d'unités). See ISO Standard ISO 80000-1:2009 [17].

7 SECONDARY ELEMENT DEFINITIONS

Type Name **(**derived from**)**	**Description**
AudienceType	A string used to identify the expected audience of the KPI. No standard types are defined. Audience types identify the specific internal audience within the company using the KPI. It may identify an organizational unit, or position within the company.
DescriptionType	A string containing a description of an element.
EffectModelType	A MIME type that specifies the effect model associated with the KPI. See *ISO 22400-1 Automation systems and integration – Key Performance Indicators (KPIs) for Manufacturing operations management — Part 1: Overview, concepts and terminology* [14] for definitions of the effect model.
ID	An identification of an exchanged element. The ID may be persistent when used in a Property, KPI Definition, KPI Instance, and KPI Value.
Name	A human readable identification of an exchanged element.
ProductionMethodologyType	Identifies the expected audience of the KPI. This may be either a standard type or an application specific extended type. Standard enumerations are: "**Batch**", "**Continuous**", "**Discrete**" and "**Other**". • **Batch** → The KPI is normally appropriate for batch production processes. • **Continuous** → The KPI is normally appropriate for continuous production processes. • **Discrete** → The KPI is normally appropriate for discrete production processes. If "**Other**" then the type is an application specific extension and the value is defined in the attribute "OtherValue".
KPIDefinitionProperty *KPIInstanceProperty* *KPIValueProperty* **PropertyType**	Defines a property of a KPI definition, instance or value. A property contains an ID and Value. Properties may be nested. Properties may contain descriptions. 

Type Name **(**derived from**)**	**Description**
RangeType	Defines a range for a KPI instance. There are no standard ranges defined, but a range will normally be the mathematical limits (0%-100%), or a nominal range (4.5 – 10.2) for the KPI. 
ResourceReferenceType	Defines a reference to a resource. The resource is identified by type and ID. See ISA 95.02 [2] for a list of standard resource types.  Example: ResourceReference/ID → 1001 ResourceReference/ResourceType → “Equipment” ResourceReference/ResourceID → “Unit 52”
ScopeType	A string that identifies the resource element that the KPI is relevant for. Example: work unit, work center or production order, product or personnel.
TrendType	Identifies how values should be interpreted. This may be either a standard type or an application specific extended type. Standard enumerations are: "**Higher-is-better**", "**Lower-is-better**", and "**Other**". • **Higher-is-better** → Higher numbers have higher value. For example, an Asset Utilization KPI. • **Lower-is-better** → Lower numbers have higher value. For example, a Quality Reject Ratio KPI. If “**Other**” then the type is an application specific extension and the value is defined in the attribute “OtherValue”.

Type Name **(**derived from**)**	**Description**
TimeRangeType	Defines either a time (StartTime or EndTime), a time range (StartTime and EndTime), or a set of intervals within a time range (StartTime, EndTime, Recurrence, and Duration).  EXAMPLE 1: If a *KPIValue* is the on-demand current *Throughput* rate then: The *EndTime* defines the time of the current value; “2014-10-08T07:59:00” EXAMPLE 2: If a *KPIValue* is the periodic *Utilization Efficiency* then: The *StartTime* defines the start of the period; “2014-09-08T08:00” The *EndTime* defines the end of the period; “2014-09-09T07:59:00” EXAMPLE 2: If a *KPIValue* is the weekly average for the first shift *OEE* for a month, then: The *StartTime* defines the start of the week; “2014-09-08T08:00” The *EndTime* defines the end of the week; “2014-10-08T07:59:00” The *Recurrence* defines once per day, “R/P1D” The *Duration* defines the time of the first shift, “PT8H” NOTE: The TimeRange element is an extension to the specification in ISO 22400-1 to handle a more complete definition of the timing associated with the KPI data and its collection.

Type Name **(**derived from**)**	**Description**
TimingType	Identifies how often the KPI should be calculated. This may be either a standard type or an application specific extended type. Standard enumerations are: "**Real-time**", "**Periodically**", "**On-demand**" and "**Other**". • **Real-time** → The KPI should be calculated after each new data acquisition event. • **Periodically** → The KPI should be calculated on a periodic basis. • **On-demand** → The KPI should be calculated on demand. If "**Other**" then the type is an application specific extension and the value is defined in the attribute "OtherValue".
UsesKPIDefinitionID **IdentifierType**	Defines the ID of another KPI Definition which is used for calculating this KPI.
UsesKPIInstanceID **IdentifierType**	Defines the ID of another KPI Instance which is used for calculating this KPI.

8 TRANSACTION DEFINITIONS

KPI-ML contains a set of elements used to support the transactions as defined in the ISA 95 Part 5 Business-to-Manufacturing Transaction [3] standard. Transactions define sets of messages that are exchanged between applications according to a specific set of rules. The transaction model follows the OAGiS 9.6 model for transaction messages using the OAGiS XML schema structure, but using data objects (nouns) that are KPI-ML elements (relating to the ISO 22400 data objects).

Transaction messages are based on the concept of VERBS and NOUNS. Verbs define the action to be taken or the response to an action. Nouns define the data objects the actions are taken on. The top level element of a XML document (the message) is named as the combination of the verb and the noun. For example, a "Get" verb on ***KPIValue*** nouns would have an element named ***GetKPIValue***.

Three different transaction models are defined:

1. A PULL model where a user of data requests the data from a provider using a GET verb, and where the provider of the data responds with a SHOW verb.
2. A PUSH model where a provider of data requests an action (processing, changing, or canceling) on the data by another user.
    a) A request to process the attached data is sent using the PROCESS verb and an optional response to the processing is returned using the ACKNOWLEDGE verb.

    Note: The definition of word "process" as meaning "deal with" or "handle". A PROCESS verb is often the equivalent of a command to add the data, but usually the receiving entity performs further actions as a result of receiving the data.

    b) A request to change information is sent using a CHANGE verb and an optional response to the change is returned using the RESPOND verb.

    c) A request to cancel the attached data is sent using the CANCEL verb.

    Note: The request to cancel indicates that the sender no longer needs the data. Because the CANCEL is not sent by the owner of the data, the data are not necessarily deleted.
3. A PUBLISH model where the publisher of data sends to users (subscribers) of the data.

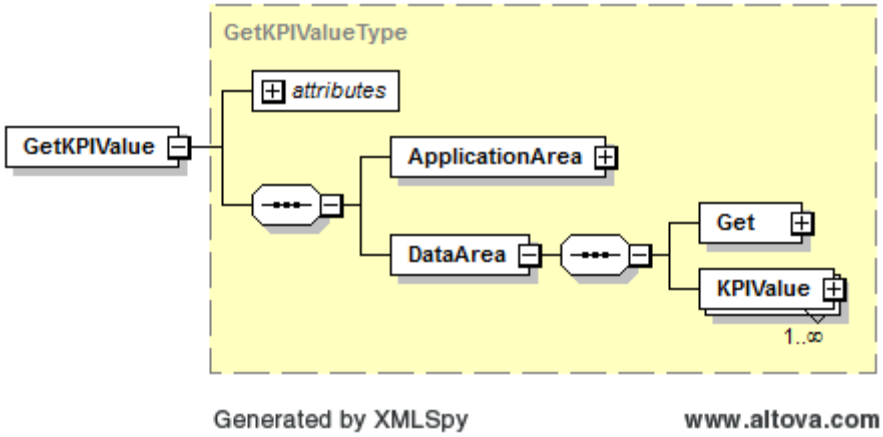
a) A notification of new data is sent using a SYNC verb and the ADD option.

b) A notification of changed data is sent using a SYNC verb and the CHANGE option.

c) A notification of deleted data is sent using a SYNC verb and the DELETE option.

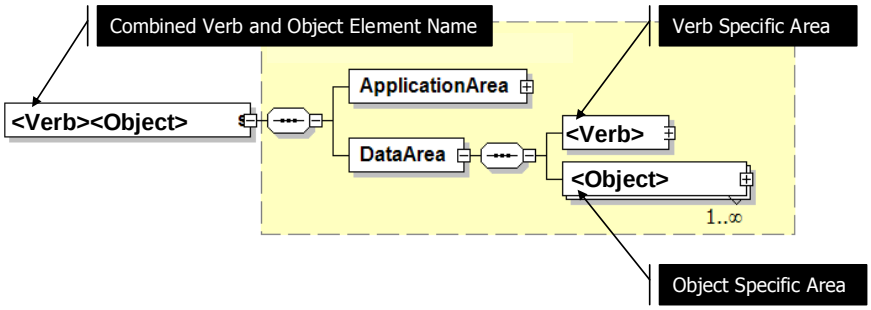
8.1 Standard Transaction Element Structure

The standard structure used for all transaction elements is an element with the verb name prefixed to the element name. For example, the element used to contain a "***Get***" message for a "***KPIValue***" element would be "***GetKPIValue***". Each transaction element contains two elements, an ***ApplicationArea*** and a ***DataArea***, as shown in the figure and partial XML sample below.




```
<GetKPIValue … releaseID="KPI-ML-V01">
   <ApplicationArea>
         …
   </ApplicationArea>
   <DataArea>
         <Get>
               …
         </Get>
         <KPIValue>
               …
         </KPIValue >
   </DataArea>
```

All transaction elements contain the same ***ApplicationArea*** element (see definition in Section 8.7). Each ***DataArea*** is unique to the specific element type being exchanged. The DataArea contains two elements, an element that is specific to the verb (***Get***, ***Show***, ***Process***, ***Confirm***, ***Acknowledge***, …) and an element that defines the specific exchanged element (***KPIDefinition***, ***KPIInstance***, and ***KPIValue***).



All common transaction element types are prefixed with "***Trans***" and postfixed with "***Type***". For example the ***ApplicationArea*** is defined in the type ***TransApplicationAreaType***, and the **Get** is defined in the type ***TransGetType***.

8.2 Message Confirmation

Any message may request confirmation of receipt of the message using a CONFIRM option that is defined in the message's *ApplicationArea*. The confirmation indicates successful processing of the message and returns error conditions if the initiating message could not be processed. The CONFIRM option may specify the following options:

Option	Option Description
Never	No confirmation requested. (*Note: Default value if option not defined.*)
OnError	Send back a confirmation only if an error has occurred.
Always	Always send a confirmation regardless of the local processing.

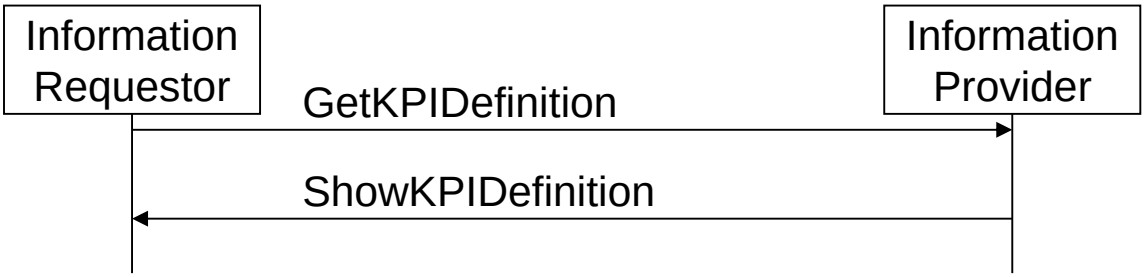
All confirmations are returned in a single message (XML element) type of *ConfirmBOD*. (Note: This follows the OAGiS definitions, where BOD is short for Business Object Document.)

***NOTE**: While any message may request confirmation (including a ConfirmBOD message), the recommended use is to only request confirmations for critical messages and only on CANCEL messages. (Confirmation on SYNC messages may lead to a large number of messages that the publisher could take no effective action on, GET messages have SHOW responses, PROCESS messages have ACKNOWLEDGE responses, and CHANGE messages have RESPOND responses.)*

8.3 PULL Transaction Model

The PULL transaction model is used when a user of data requests information from a provider of the data. The request is defined in a message that contains a GET verb and an empty or partially defined element.

For example the following diagram indicates a GET/SHOW transaction.



The ISO 22400 standards do not, at this point in time, define transaction rules for GET/SHOW data exchanges. The following tables define recommended rules for interpreting GET transaction messages.

Table 1 - GET rules for KPI Definitions

Value of KPI Definition ID	Action on Object(s) Specified for the GET verb on KPI Definitions
IDs specified	Defines a request that the receiver is to return, in a SHOW message, all attributes about the specified *KPI Definition*, all properties and their attributes, and the IDs of *KPI Instances* that are associated with the *KPI Definition*.
Wildcard specified	Defines a request that the receiver is to return, in a SHOW message, all attributes and properties about the *KPI Definitions* that match the wildcard ID and the IDs of *KPI Instances* that are associated with the *KPI Definitions*. To return all *KPI Definitions*, specify a "*" as the wildcard.

Table 2 - GET rules for KPI Instances

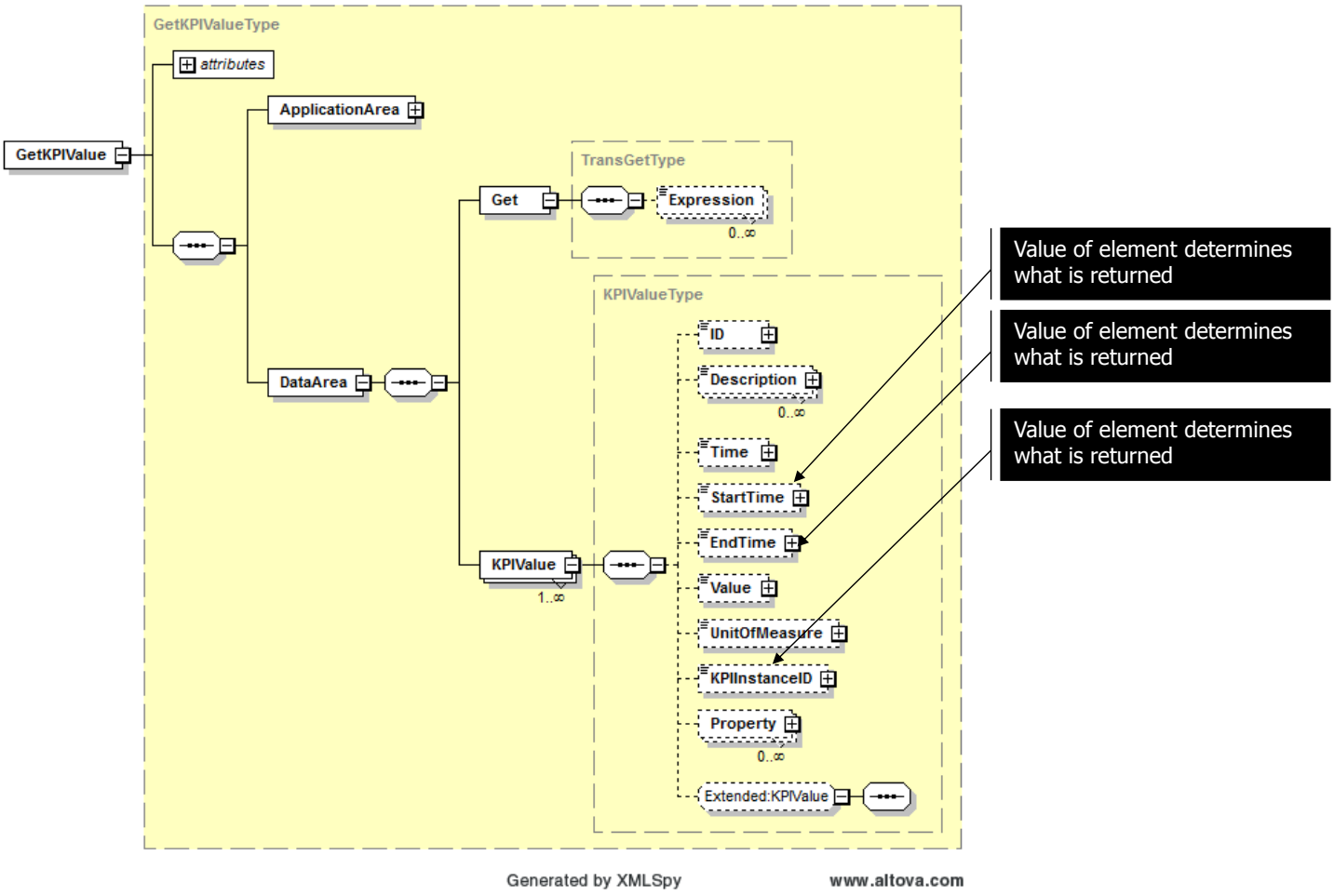
Value of KPI Instance ID	Action on Object(s) Specified for the GET verb on KPI Instance
IDs specified	Defines a request that the receiver is to return, in a SHOW message, all attributes about the specified *KPI Instance*, all properties and their attributes.
Wildcard specified	Defines a request that the receiver is to return, in a SHOW message, all attributes and properties about the *KPI Instances* that match the wildcard. To return all *KPI Definitions*, specify a "*" as the wildcard.

Table 3 - GET rules for KPI Values

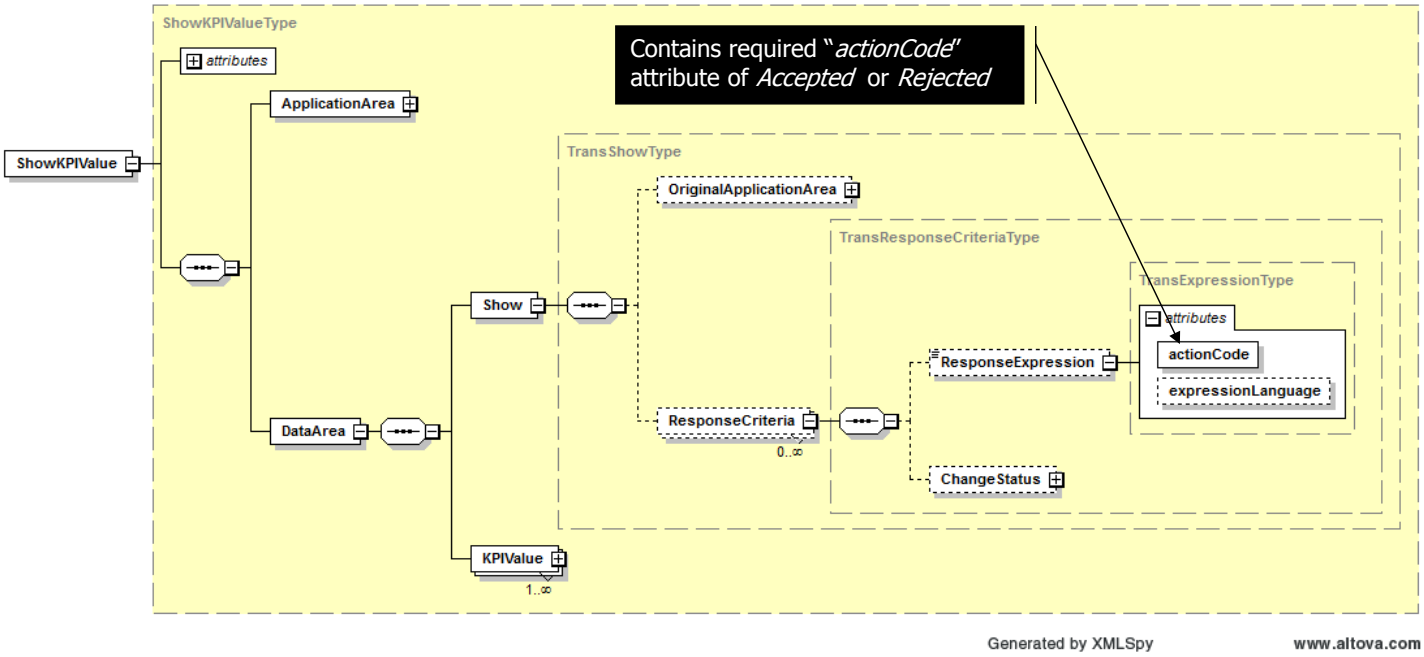
KPI Value Element	Action on Object(s) Specified for the GET verb on KPI Definitions
KPI Instance ID	Defines a request that the receiver is to return, in a SHOW message, all *KPI Values* for the specified *KPI Instance ID*, in conjunction with the other GET rules.
Start Time	Defines a request that the receiver is to return, in a SHOW message: ▪ For **periodic** KPIs - all *KPI Values* that have a *Start Time* greater than or equal to the specified start time, in conjunction with the other GET rules. ▪ For **on demand** KPIs – the current KPI Value, in conjunction with the other GET rules. ▪ For **real-time** KPIs – all KPI Values that have a *Start Time* greater than or equal to the specified start time, in conjunction with the other GET rules.
End Time	Defines a request that the receiver is to return, in a SHOW message, : ▪ For **periodic** KPIs - all *KPI Values* that have an *End Time* less than or equal to the specified end time, in conjunction with the other GET rules. ▪ For **on demand** KPIs – the current KPI Value, in conjunction with the other GET rules. ▪ For **real-time** KPIs – all KPI Values that have an *End Time* less than or equal to the specified end time, in conjunction with the other GET rules.

Periodic KPI Example:

- GetKPIValue
  - KPIValue/KPIInstanceID → OEE
  - KPIValue/StartTime → 2014-08-06 8:00.00
  - KPIValue/EndTime → 2014-08-07 7:59.59
- Returns
  - All OEE KPI Values from 2014-08-06 8:00.00 to 2014-08-07 7:59.59



The SHOW message contains a required response *actionCode* attribute of either *Accepted* or *Rejected* in the *Show/ResponseCriteria/ResponseExpression* element as shown in the figure below.



8.4 PUSH Transaction Model

The PUSH model uses PROCESS/ACKNOWLEDGE, CHANGE/RESPOND, and CANCEL messages for an application that is not the owner of data to request processing, changing, or canceling data to the data owner.

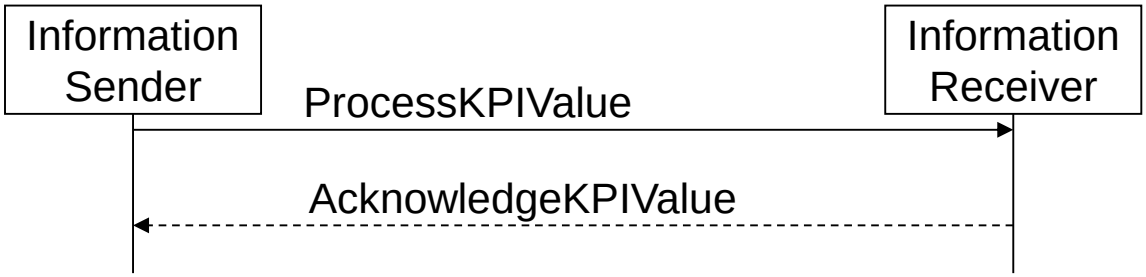
8.4.1 Transactions

For example, the following diagram indicates a Process/Acknowledge transaction for a ***KPIValue*** element. The PROCESS message may contain a CONFIRM option and an ACKNOWLEDGE option, but normally only the ACKNOWLEDGE would be specified (not both).

The PROCESS Acknowledgement option may specify the following options:

Option	Option Description
Never	No acknowledge requested. (*Note: Default value if option not defined.*)
Always	Always send an acknowledge response.

See the ISA 95 Part 5 standard for a complete definition of the action to be performed as the result of receiving a PROCESS message for each object (element) type.

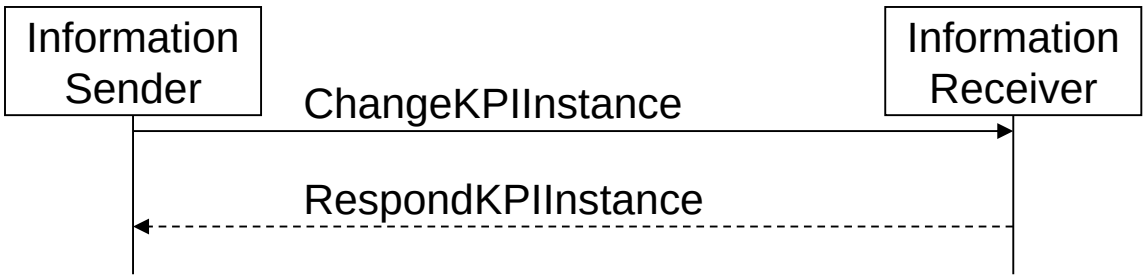


The following diagram illustrates a CHANGE/RESPOND transaction for a ***ProductDefinition*** element. The CHANGE message may contain a CONFIRM option and a RESPOND option, but normally only one or neither would be specified (not both).

The CHANGE respond option may specify the following options:

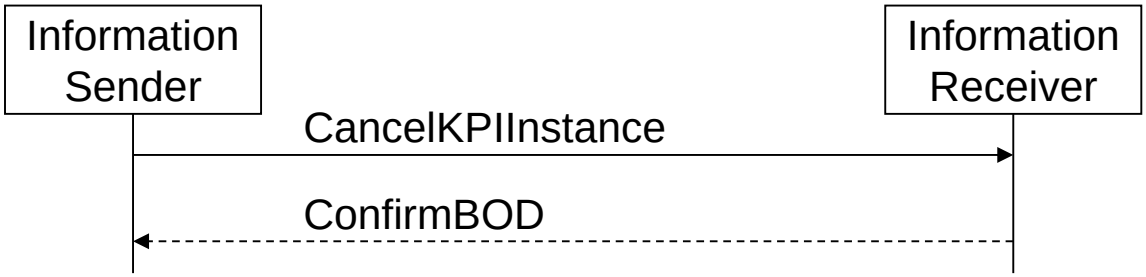
Option	Option Description
Never	No response requested. (*Note: Default value if option not defined.*)
Always	Always send a response.

See the ISA 95 Part 5 standard for a complete definition of the action to be performed as the result of receiving a CHANGE message for each object (element) type.



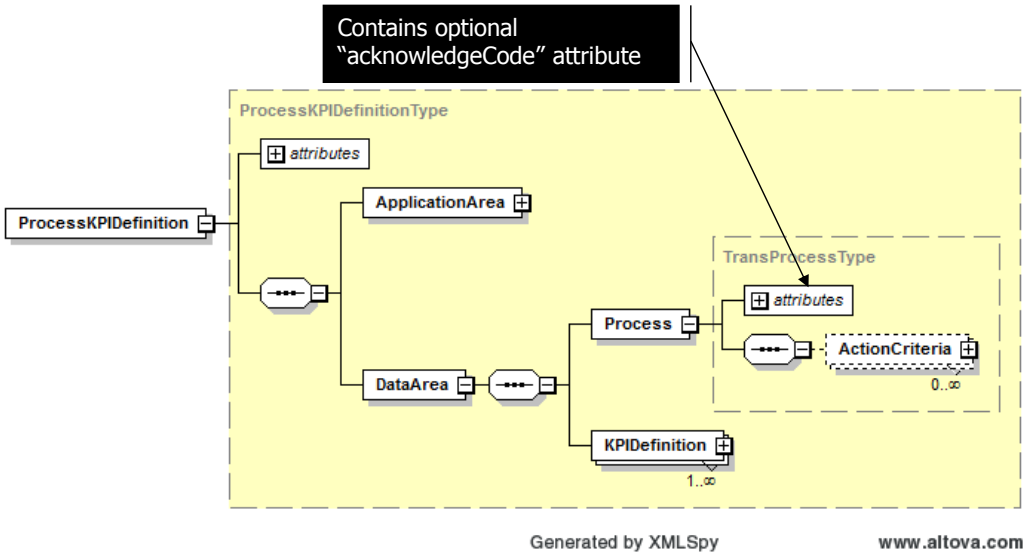
The following diagram illustrates a CANCEL transaction for a ***KPIInstance*** element. The CANCEL message may contain a confirmation option.

See the ISA 95 Part 5 standard for a complete definition of the action to be performed as the result of receiving a CANCEL message for each object (element) type.



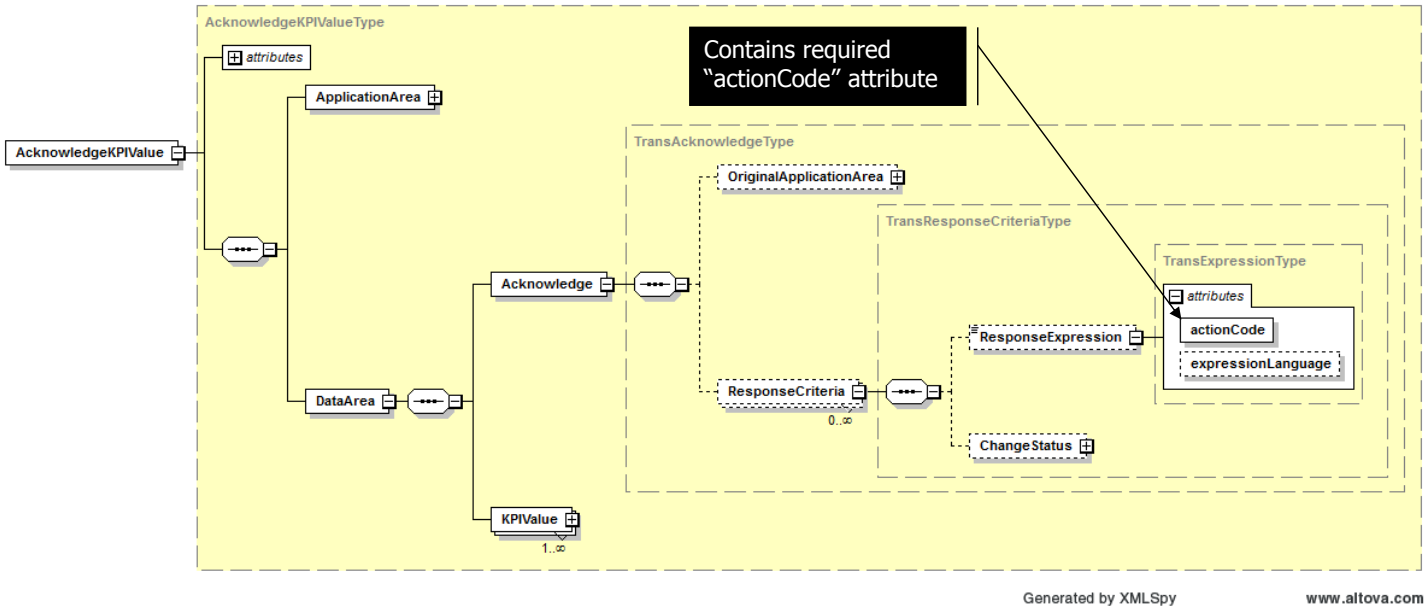
8.4.2 Process Acknowledgment

The acknowledge option is defined using optional attributes in PROCESS messages. The PROCESS verb contains an optional *acknowledgeCode* attribute, as shown below in the *Process* element.



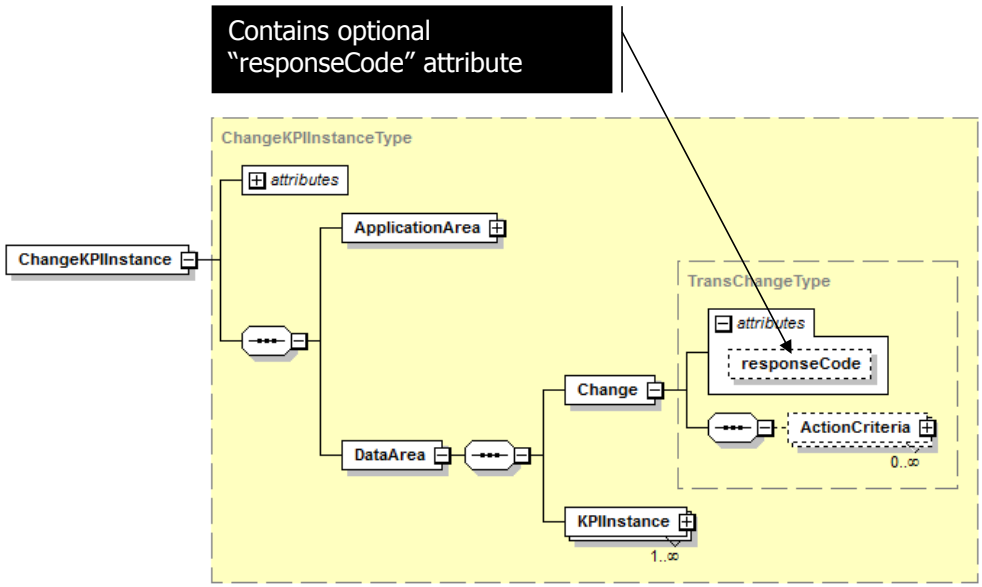
For consistency with OAGiS verb definitions, the PROCESS verb also contains an *actionCode* attribute with *ActionExpression* elements. This element is not used in the KPI-ML transactions.

The ACKNOWLEDGE message contains a required status attribute in the *Acknowledge/ResponseCriteria/ResponseExpression* element, as shown below.



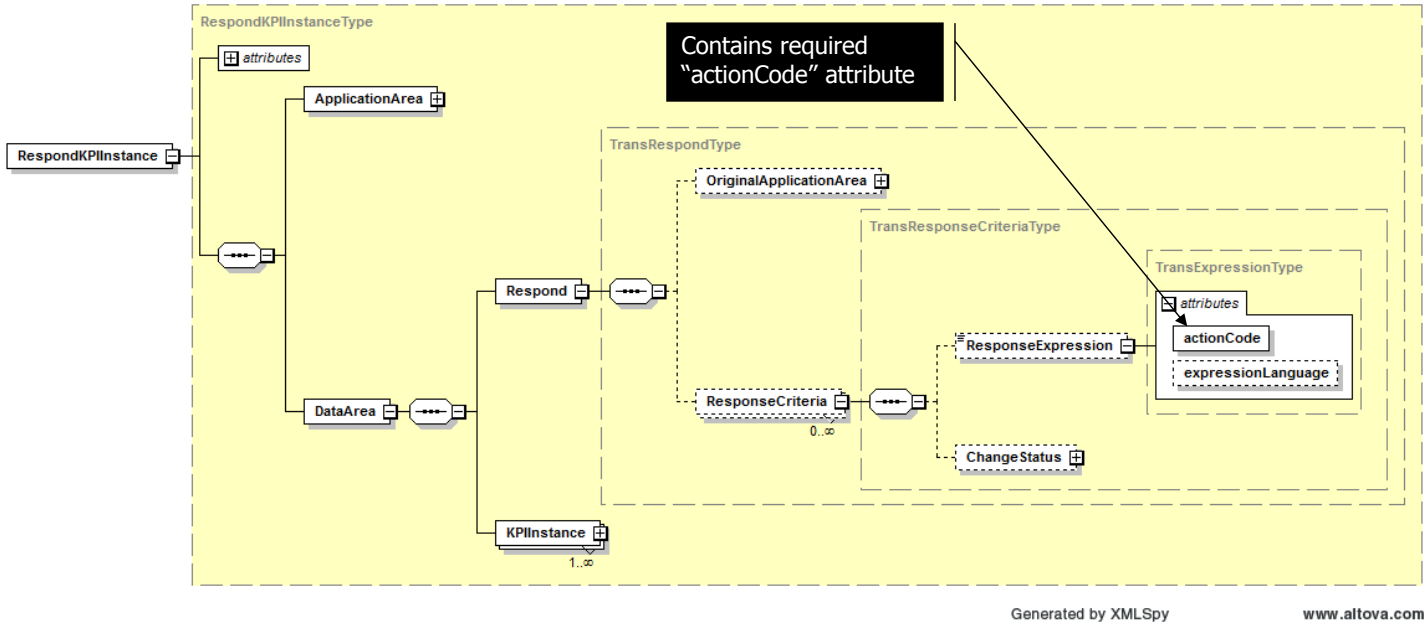
8.4.3 Change Response

The CHANGE verb contains an optional *responseCode* attribute, as shown below in the *Change* element.



For consistency with OAGiS verb definitions, the CHANGE verb also contains an *actionCode* attribute with ActionExpression elements. This element is not used in the KPI-ML transactions.

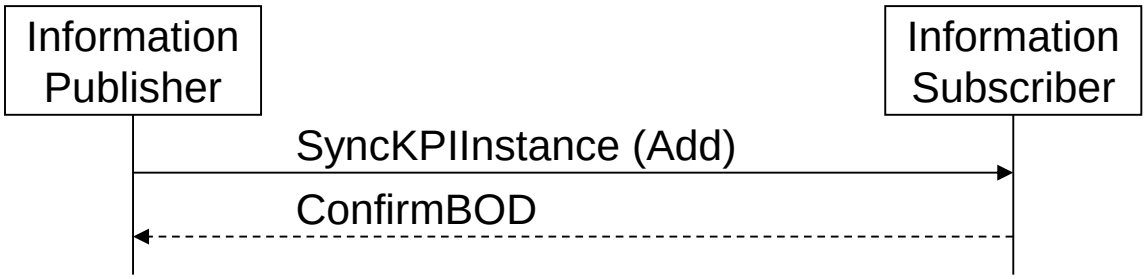
The RESPOND message contains a required *actionCode* attribute in the *Respond/ResponseCriteria/ResponseExpression* element.



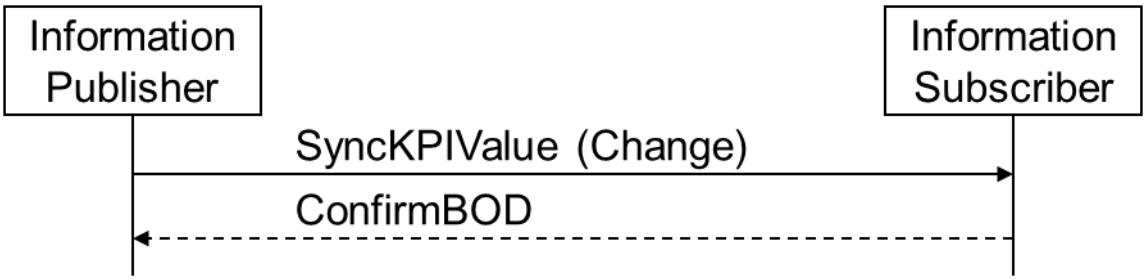
8.5 PUBLISH Transaction Model

The PUBLISH model uses SYNC messages with ADD, CHANGE, or DELETE options to indicate the action to be taken on the published data.

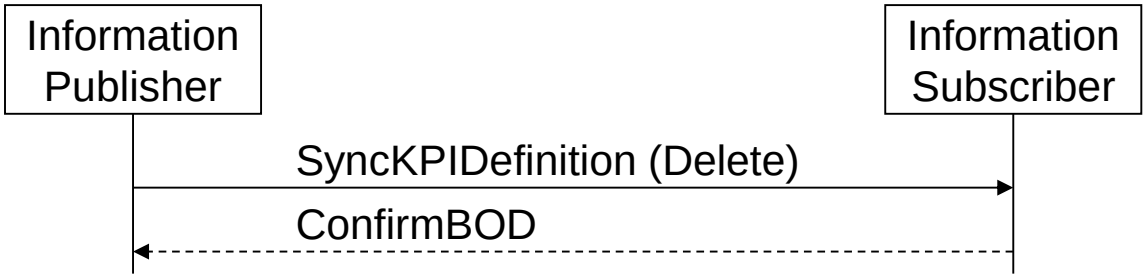
The following diagram illustrates a SYNC ADD transaction for a ***KPIInstance*** element. The ***SyncKPIInstance*** (Add) message may be sent to multiple information subscribers when a new ***KPIInstance*** object is defined, but only one is shown in the example. The SYNC message may contain a confirmation option, but this is not normally used in PUBLISH transactions. See the ISA 95 Part 5 standard for a complete definition of the action to be performed as the result of receiving a SYNC ADD message for each object (element) type.



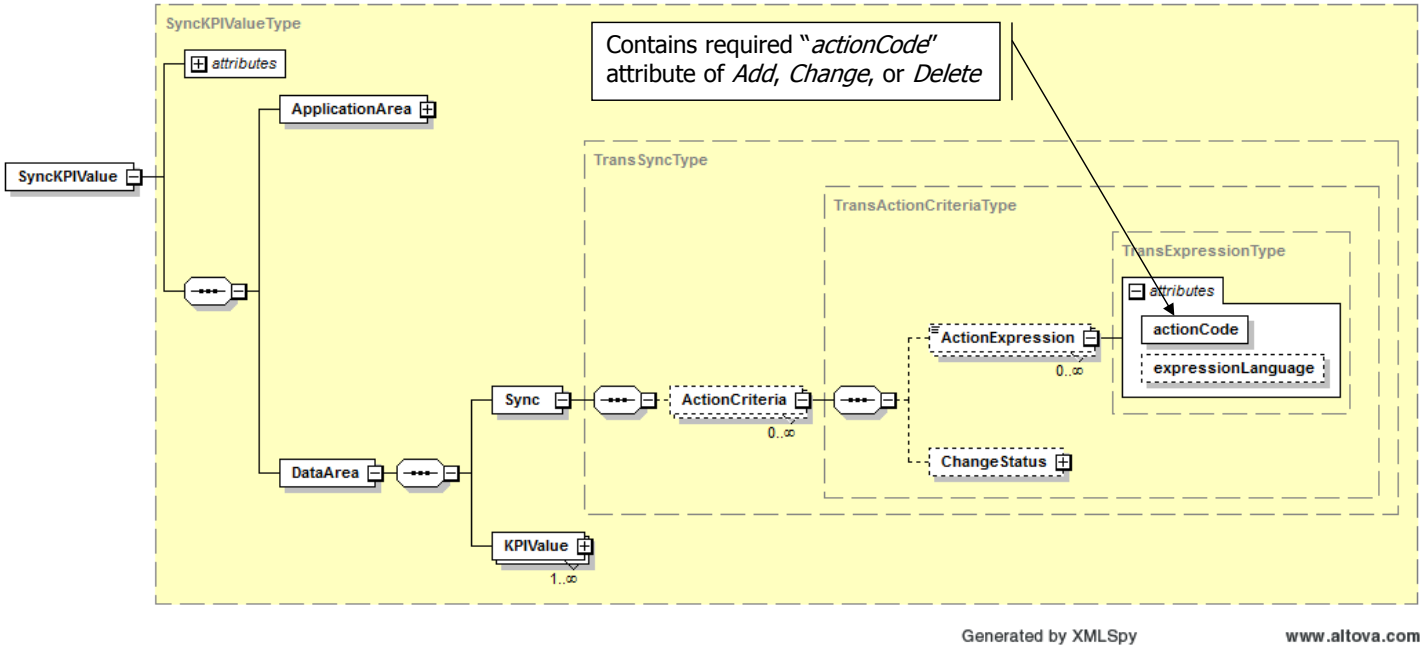
The following diagram illustrates a SYNC CHANGE transaction for a ***KPIValue*** element. The SyncKPIValue (Change) message may be sent to multiple information subscribers when a ***KPIValue*** object is changed, but only one is shown in the example. The SYNC message may contain a confirmation option, but this is not normally used in PUBLISH transactions. See the ISA 95 Part 5 standard for a complete definition of the action to be performed as the result of receiving a SYNC CHANGE message for each object (element) type.



The following diagram illustrates a SYNC DELETE transaction for a ***KPIDefinition*** element. The SyncKPIDefinition (Change) message may be sent to multiple information subscribers when a ***KPIDefinition*** is deleted, but only one is shown in the example. The SYNC message may contain a confirmation option, but this is not normally used in PUBLISH transactions. See the ISA 95 Part 5 standard for a complete definition of the action to be performed as the result of receiving a SYNC DELETE message.

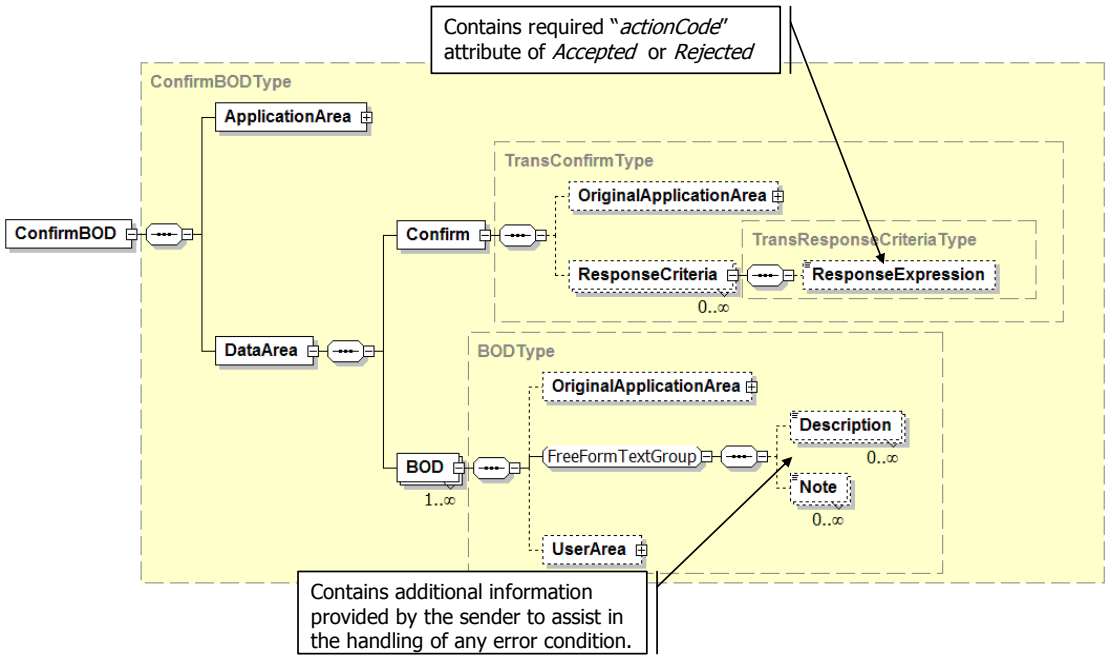


The SYNC options are specified in the required attribute of the *Sync/ActionCriteria/ActionExpression* element. There are multiple *ActionCriteria* elements allowed by the schema, but only one is used in the KPI-ML transactions.

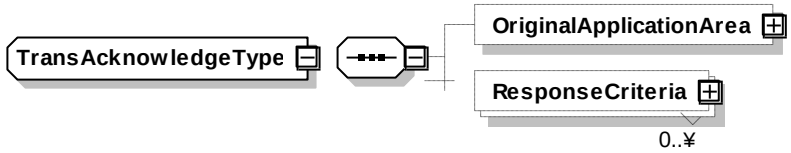


8.6 ConfirmBOD

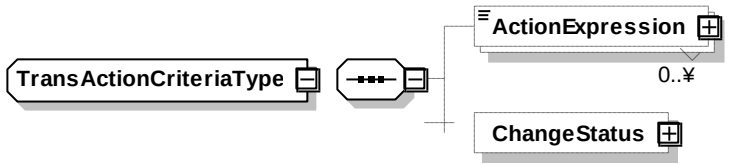
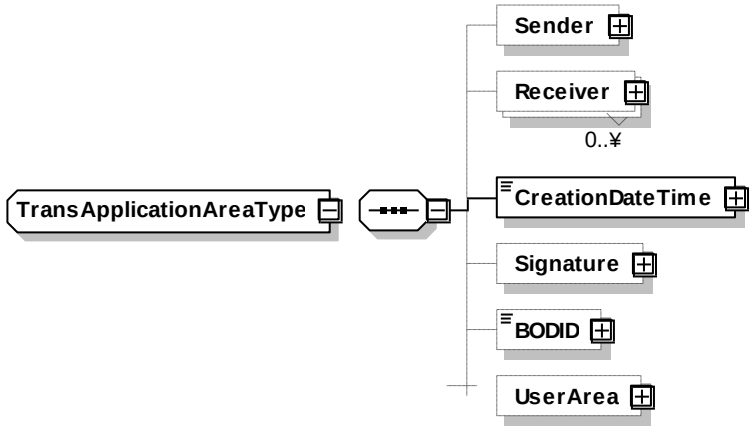
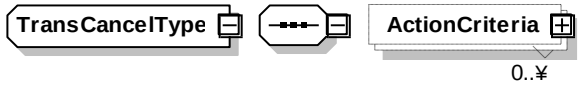
The ConfirmBOD is the message that is returned when the confirmation option is specified in a message. The actual status of the response is contained in the **actionCode** attribute in the **Confirm.ResponseCriteria.ResponseExpression**. Addition information may be contained in the free format text of the **Description** and **Note** areas in the **BOD**. The original ApplicationArea can also be returned in the ConfirmBOD for the receiver to determine source of the confirmed message.

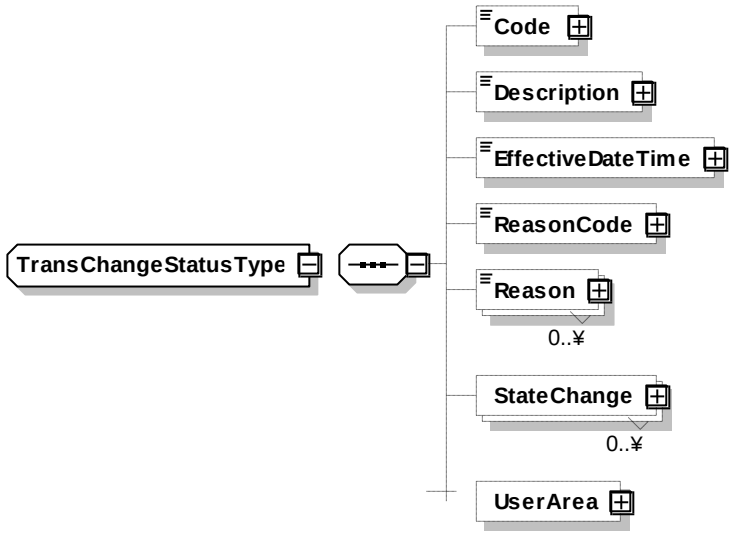
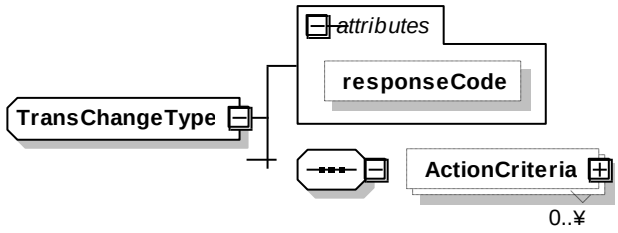


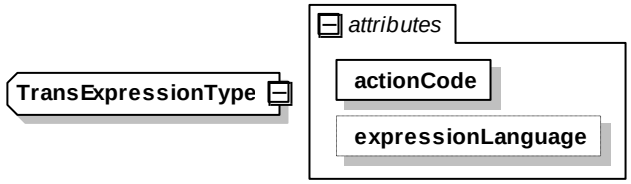
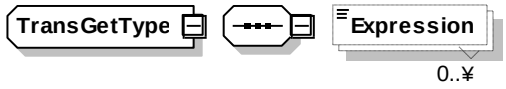
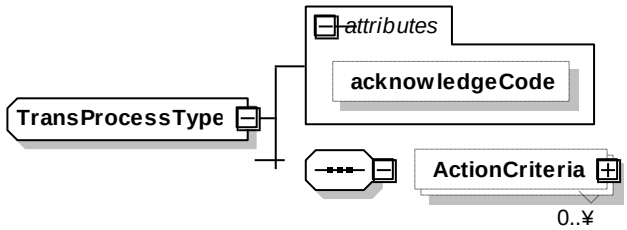
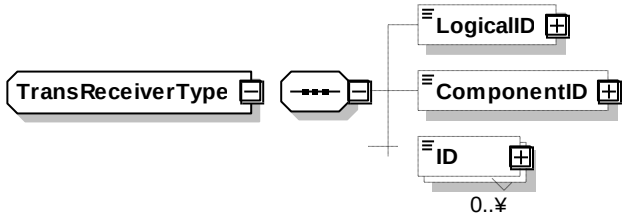
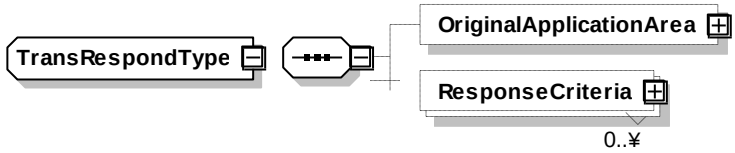
8.7 Common Transaction Elements

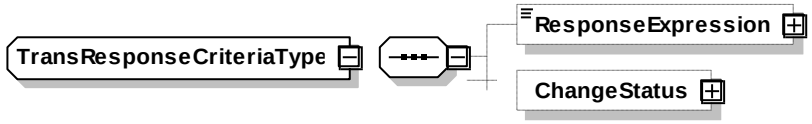
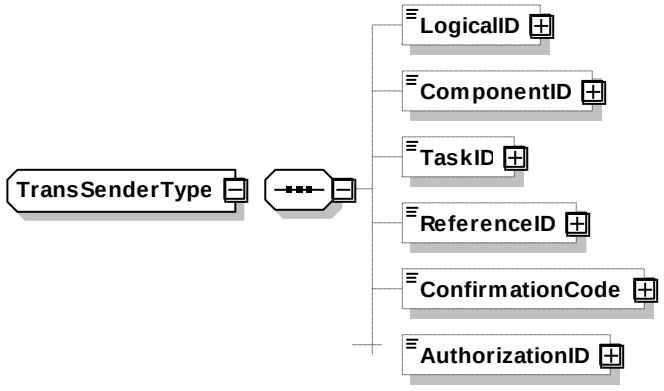
Type Name Element Name	Description
TransAcknowledgeType Acknowledge	Complex data type for an ACKNOWLEDGE verb in an *Acknowledge<Object>* message. A complex type that contains two elements: • **OriginalApplicationArea:** An optional copy of the ApplicationArea from the requesting Process message (see ***TransApplicationAreaType***). This is included to assist in error handling by the requesting application. • **ResponseCriteria:** Zero or more elements (See ***TransResponseCriteriaType***) that contain additional acknowledge information (including an action code). o If no **ResponseCriteria** is present, then the action code of "**Accepted**" is the default. o The first **ResponseCriteria** element defines the acknowledgement option. The meanings of any additional **ResponseCriteria** are not defined in KPI-ML. 

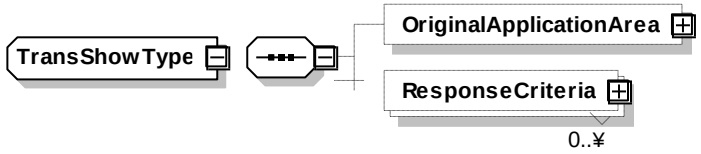
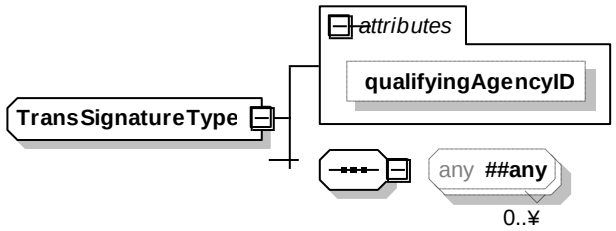
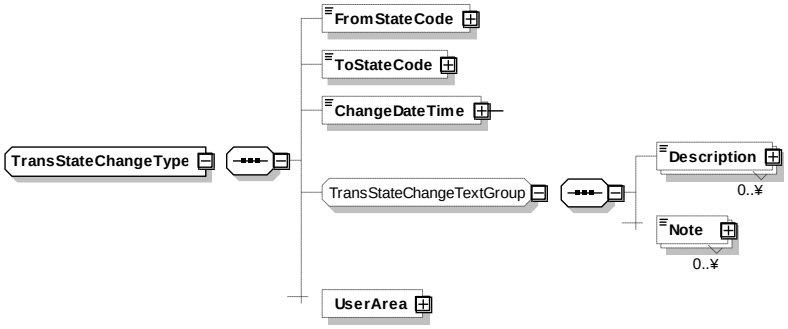
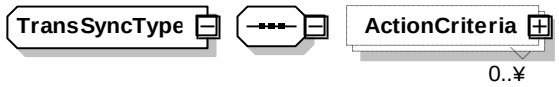
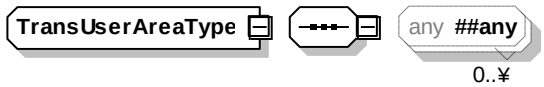
<table>
<tr>
<td>TransActionCodeEnumerationType</td>
<td>A string that contains an identification of the action to be performed as part of a verb. The action codes are used for SYNC messages and as responses in CHANGE, CONFIRM, ACKNOWLEDGE and RESPOND messages. The following enumerations are defined:
<table>
<tr><th>Enumeration</th><th>Action Meaning</th></tr>
<tr><td>Add</td><td>Used in a SYNC message to indicate a SYNC ADD action to be performed.</td></tr>
<tr><td>Change</td><td>Used in a SYNC message to indicate a SYNC CHANGE action to be performed.</td></tr>
<tr><td>Delete</td><td>Used in a SYNC message to indicate a SYNC DELETE action to be performed.</td></tr>
<tr><td>Replaced</td><td>Used in a RESPOND message to indicate that the change was performed</td></tr>
<tr><td>Accepted</td><td>Used in an ACKNOWLEDGE message to indicate that the PROCESS was performed
Used in a CONFIRM message to indicate that the action was performed.
Used in a SHOW message to indicate that the GET was performed and all data was returned. This may include a SHOW message with no data elements, if the section criteria identified no data to be returned.</td></tr>
<tr><td>Modified</td><td>Used in an ACKNOWLEDGE message to indicate that the PROCESS was performed and that the information was modified.
Used in a RESPOND message to indicate that the CHANGE was performed and that the information was modified.
Not used in a SHOW message.</td></tr>
<tr><td>Rejected</td><td>Used in an ACKNOWLEDGE message to indicate that the PROCESS was rejected.
Used in a RESPOND message to indicate that the CHANGE was rejected.
Used in a CONFIRM message to indicate that the action was rejected.
Used in a SHOW message to indicate that the GET was not performed and that no data was returned.</td></tr>
</table>
</td>
</tr>
<tr>
<td>TransActionCodeType</td>
<td>A union type of an enumeration (see TransActionCodeEnumerationType) and a normalizedString. This allows either a defined enumeration value (see above) or a user defined string. The meanings of any user defined action code strings are not defined in KPI-ML.</td>
</tr>
</table>

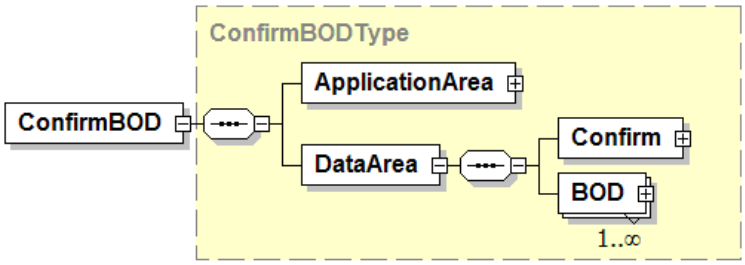
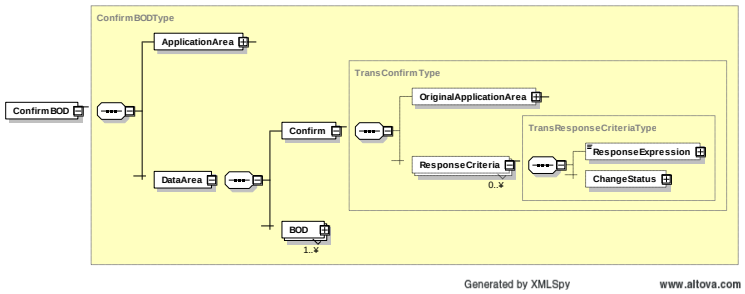
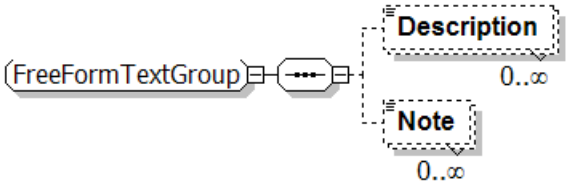
TransActionCriteriaType ActionCriteria	Data Type for a SYNC, PROCESS, CHANGE, and CANCEL message. It contains one optional element **ActionExpression** (see ***TransExpressionType***) that contains an action code for SYNC messages. It also contains an optional **ChangeStatus** (see **TransChangeStatusType**) element for a definition of the change. If no **ActionExpression** is defined for a SYNC message, then the action code of "**Add**" is the default. 
TransApplicationAreaType ApplicationArea OriginalApplicationArea	A complex type that contains: • An optional identification of the sender of the message (see **TransSenderType**) • Zero or more optional identifications of the receiver of the message (see **TransReceiverType**) • A required element with the creation date & time of the message. • An optional electronic signature that can be used to sign the transaction message • An optional ID (**BODID**) to be applied to exchanged data object. This should be a GUID (Globally Unique Identifier) that uniquely identifies the data object. • An optional user area for user extended data. 
TransCancelType	Data type for a CANCEL verb in a *Cancel<Object>* message. Contains an optional attribute **responseCode** of type **TransResponseCodeType**. The responseCode specifies if a response is required. The complex type also contains an optional ActionCriteria (see **TransActionCriteriaType**) element for compatibility with OAGiS; however the ActionCriteria elements are not defined in a **TransCancelType** element in KPI-ML. 

TransChangeStatusType ChangeStatus	Defines the description for a response. **Code** (CodeType): A user defined element for communication of all codes. No standard code types are defined. **Description** (DescriptionType): A text description of the overall reason for the response. **EffectiveDateTime** (DateTimeType): The effective date and time that response was generated, to allow backtracking of the reason for the response. **ReasonCode** (CodeType): Identifies the reason for the response activity. **Reason** (TextType): Text description of the reasons for the response. **StateChange**: Information about any state changes associated with the response. **UserArea**: User defined ##any type. 
TransChangeType	Data type for a CHANGE verb in a *Change<Object>* message. Contains an optional attribute **responseCode** of type **TransResponseCodeType**. The responseCode specifies if a response is required. The complex type also contains an optional ActionCriteria (see **TransActionCriteriaType**) element for compatibility with OAGiS; however the ActionCriteria elements are not defined in a **TransChangeType** element in KPI-ML. 
TransConfirmationCodeType	A string used to indicate a confirmation, acknowledge, or respond code option in a message. The value must be one of the following standard enumerations: **Always**, **Never**, **OnError** • Always → Always return a confirm, acknowledge, or respond message. • Never → Never return a confirm, acknowledge, or respond message. • OnError → Only return a confirm, acknowledge, or respond message if an error occurred during processing of the message's action.

TransExpressionType ActionExpression ResponseExpression	A complex type with: • A simple type of "**token**" • A required attribute of **actionCode** (see **TransActionCodeType**) • An optional **expressionLanguage** attribute that specifies the language that may be used to interpret the tokens in the expression. The meaning of any text included in the token in a **TransExpressionType** is not defined in KPI-ML. 
TransGetType Get	Data type for a GET verb in a *Get<Object>* message. There are no attributes. 
TransProcessType Process	Data type for a PROCESS verb in a *Process<Object>* message. Contains an optional attribute **acknowledgeCode** of type **TransResponseCodeType**. The responseCode specifies if a response is required. 
TransReceiverType Receiver	Contains information about the expected receiver of the message. This contains an optional **LogicalID** of the server and application for which the BOD is intended. This contains an optional **ComponentID** of the server and application for which the BOD is intended. It provides a finer level in addition to the LogicalID. This contains zero or more optional **ID**s for the receiver of the message. 
TransRespondType Respond	Data type for a RESPOND verb in a *Respond<Object>* message. 

TransResponseCodeType	A string used to indicate if response is requested for the sent. The value must be one of the following standard enumerations: **Always**, **Never** • Always → Always return a response message. • Never → Never return a response message.
TransResponseCriteriaType ResponseCriteria	Data Type for an ACKNOWLEDGE, CONFIRM, SHOW, and RESPOND response. It contains one optional element **ResponseExpression** (see ***TransExpressionType***) that contains an action code. If no **ResponseExpression** is defined, then the action code of "**Accepted**" is the default. **ChangeStatus** is an optional element that contains the reason for the response. 
TransSenderType	A complex type that identifies characteristics and control identifiers that relate to the application that created the transaction message. The sender area can indicate the logical location of the application and/or database server, the application, and the task that was executing to create the data object. • The format for the **LogicalID** is not defined in KPI-ML. It may contain a logical (not physical) identification of the sending task. • The format for the **ComponentID** is not defined in KPI-ML. It may contain additional detail for the **LogicalID**. • The format for the **TaskID** is not defined in KPI-ML. It may describe the business event that initiated the need for the Business Object Document to be created. • The format for the **ReferenceID** is not defined in KPI-ML. It may contain additional information that enables the sending application to indicate the instance identifier of the event or task that caused the data to be created. • The format for **ConfirmationCode** is defined in **TransConfirmationCodeType**. • The format for the **AuthorizationID** is not defined in KPI-ML. It may identify the authorization level of the user or application that is sending the data. This authorization level may indicate to the receiving system what can be done on request. 

TransShowType	Data type for a SHOW messages. 
TransSignatureType	A ##**any** type that is used if the message is to be signed. It supports any digital signature that may be used by an implementation. The optional qualifyingAgencyID attribute identifies the agency that provided the format for the signature. In order to support digital signature specifications currently available or that will be developed in the future, the Signature element is defined to have an *##any* content from other namespaces. The choice of which digital signature to use is left up to the specific implementation. 
TransStateChangeType StateChange	Defines any state change associated with the response, such as a change from effective to obsolete. • **FromStateChange** (CodeType): Old state • **ToStateChange** (CodeType): New state • **ChangeDateTime** (DateTimeType): Date and time the change occurred. • **Description** (TextType): Descriptions of the change. • **Note** (TextType): Secondary notes associated with the change. • **UserArea**: User ##any type 
TransSyncType	Data type for SYNC messages. 
TransUserAreaType UserArea	A ##**any** type that is used to contain user data in the application area. 

ConfirmBOD	The ConfirmBOD is a complex type that contains an application area and data area. The data area contains a Confirm element and a BOD element. 
TransConfirmType	The **TransConfirmType** contains a copy of the original application area sent on the message requesting confirmation. It also contains a **ResponseCriteria** that contains a **ResponseExpression** that contains an **actionCode** attribute with the confirmation status. 
BOD	A general type that is used to contain additional return status information. 
FreeFormTextGroup	A group of Descriptions and Notes that are related and contain additional return status information. 

8.8 KPI-ML and OAGiS Differences

KPI-ML is designed to implement a subset of the OAGiS 9.6 message rules that are consistent with the ISA 95 Part 5 definitions. Specifically the KPI-ML ***ApplicationArea*** and verb elements are subsets of the full OAGiS 9.6 specifications, with the differences listed in the following table.

Element	Differences
Show	The OAGiS 9.6 "*Show*" specification includes a set of optional attributes (recordSetStartNumber, recordSetCount, recordSetTotal, recordSetCompleteIndicator, and recordSetReferenceId) that are used by the responding task to indicate the status of the request and to define the scope of the information returned. These attributes are not defined for KPI-ML. The Show message should return all elements of the Get request.
Get	The OAGiS "Get" specification includes a set of optional attributes (uniqueIndicator, maxItems, recordSetSaveIndicator, recordSetStartNumber, and recordSetReferenceId) that are used by the requesting application to control how many elements are returned. These attributes are not defined for KPI-ML. The Show message should return all elements of the Get request.
ActionExpression **ResponseExpression**	In KPI-ML this contains a required *actionCode* attribute that contains an *Accepted* or *Rejected* value.
ConfirmBOD	KPI-ML contains only the Original Application Area, a free form text group, and user data. OAGiS 9.6 also contains BOD Failure Message, BOD Success Message, and Partial BOD Failure Message areas.
DateTimeType	KPI-ML has the DateTimeType derived from the xsd:dateTime type. OAGiS 9.6 has the DateTimeType derived from xsd:string. KPI-ML is more restrictive than OAGiS, but OAGiS recommends the use of ISO 8601 CE format.

9 KPI DEFINITION EXAMPLE

An example of a KPI definition from ISA 22400-2 is:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 rel. 3 sp1 (http://www.altova.com)-->
<KPIDefinition
      xsi:schemaLocation=http://www.mesa.org/xml/KPI-ML-V01 KPI-ML-V01.xsd
      xmlns="http://www.mesa.org/xml/KPI-ML-V01"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <ID>OLR100</ID>
      <Description> The Other Loss ratio is the relationship of the
          quantity of loss not related to production, storage or
          transportation (OL) to the quantity of Consumed Material (CM).
      </Description>
      <Name>Other Loss Ratio</Name>
      <Scope> Work unit </Scope>
      <Scope> defect type </Scope>
      <Formula>Other loss ratio = OL/CM</Formula>
      <UnitOfMeasure>%</UnitOfMeasure>
      <Range>
            <ID>Natural</ID>
            <Description>Natural Range</Description>
            <LowerLimit>0</LowerLimit>
            <UpperLimit>100</UpperLimit>
      </Range>
      <Trend>Lower-is-better</Trend>
      <Timing>On-demand</Timing>
      <Timing>Periodically</Timing>
      <Timing>Real-time</Timing>
      <Audience>Operator</Audience>
      <Audience>Supervisor</Audience>
      <Audience>Management</Audience>
      <ProductionMethodology>Batch</ProductionMethodology>
      <ProductionMethodology>Continuous</ProductionMethodology>
      <Notes>” The other loss ratio evaluates losses that have not occurred
            during production, storage, or transportation.
            See also production loss ratio”
      </Notes>
</KPIDefinition>
```

10 SCHEMA EXTENSIONS

10.1 User Enumeration Extensibility

The ISO 224000-1 standard defines a set of specific enumerations or elements. KPI-ML provides support for these enumerations and adds the possibility for user extensions. The extended enumeration values are not defined in this standard and should not be considered understandable between applications without prior agreement.

All types that contain enumerated values have an enumeration value of “Other” and an attribute of “OtherValue”. Any element, in an instance document, with an extended enumeration should use the enumeration value of “Other” and place the actual enumeration in the “OtherValue” attribute.

The enumerations and “OtherValue” attribute are added in two steps in the schemas. This is done due to a restriction in W3C schemas that prevent restrictions (enumeration values) and extensions (adding an attribute) at the same time. The complex type naming convention used is a “1” in the name of temporary complex type (complex type name = ‘element name’ + ‘1’ + ‘Type’) and the same name without the ‘1’ for the final complex type name. The two step process can be ignored by XML authors because the temporary type is not intended for use in XML documents.

Schema definition:

```
  <xsd:complexType name="ProductionMethodology1Type">
       <xsd:simpleContent>
           <xsd:restriction base="CodeType">
               <xsd:enumeration value="Batch"/>
               <xsd:enumeration value="Continuous"/>
               <xsd:enumeration value="Discrete"/>
           </xsd:restriction>
       </xsd:simpleContent>
   </xsd:complexType>

   <xsd:complexType name="ProductionMethodologyType">
       <xsd:simpleContent>
           <xsd:extension base="ProductionMethodology1Type">
               <xsd:attribute name="OtherValue" type="xsd:string"/>
           </xsd:extension>
       </xsd:simpleContent>
   </xsd:complexType>>
 </xsd:complexType>
```

Used in XML document:

```
<kpiml:ProductionMethodology>
   Batch
</kpiml:ProductionMethodology >


<kpiml:ProductionMethodology OtherValue = "Mixed">
   Other
</kpiml:ProductionMethodology >
```

10.2 Extension using the Extended Namespace

Each complex type that is not an enumerated set contains an **xsd:group** reference. The reference is to the element name in the "Extended" name space. The extended namespace is defined in the user editable **KPI-ML-V02-extensions.xsd** file. The user extensions may be the specific schema extensions, or the extensions file could import company specific extensions.

Basically the end user may edit the **KPI-ML-V02-extensions.xsd** file to contain the added elements. It may include company specific extensions or vendor supplied extensions.

For example: The following schema segment defines the "Extended:Range" element within a "RangeType".

```
<xsd:schema    targetNamespace    = "http://www.mesa.org/xml/kpi-ml-v02"
               xmlns              = "http://www.mesa.org/xml/kpi-ml-v02"
               xmlns:Extended     = "http://www.mesa.org/xml/kpi-ml-v02-extensions"
               xmlns:xsd      = "http://www.w3.org/2001/XMLSchema"
               elementFormDefault = "qualified"
               attributeFormDefault   = "unqualified">

<xsd:complexType name="RangeType">
   <xsd:sequence>
      <xsd:element name="ID"               type="IdentifierType"/>
      <xsd:element name="Description"      type="DescriptionType" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="LowerLimit"       type="MeasureType" nillable="true"/>
      <xsd:element name="UpperLimit"       type="MeasureType" nillable="true"/>
      <xsd:group   ref="Extended:Range"    minOccurs="0" maxOccurs="1"/>
   </xsd:sequence>
</xsd:complexType>
```

The following schema segment defines a sample user extension to **Range** and includes two company specific extensions:

```
<xsd:group name="Range">
    <xsd:sequence>
        <xsd:element name="RangeClass"           type="xsd:string" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Accountability"       type="xsd:string" minOccurs="0"/>
        <xsd:group ref="AAA-Control:PhysLimit"   minOccurs="0" maxOccurs="1"/>
        <xsd:group ref="XYZ-Eng:LogLimit"        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:group>
```

NOTE: When company specific extensions are added to an extended element, then the extended names spaces should be added in alphabetical order with a minOccurs of 0, and all end users extended elements should be listed first. This allows the mixing of schemas from different vendors, so that any B2MML document may have none, some, or all of the extensions without any element sequencing problems.

The use of the extended namespace construct allows for intra-vendor or intra-company extensions to the schemas with full schema validation, extending the number of places that they may be applied. The extended namespace does this without affecting the ability to perform inter-vendor or inter-company information exchange as defined by the standards for exchanged information in ISO 22400-1. However, extensive use of extensions limits the interoperability of KPI-ML data exchanges.

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, BatchML, and KPI-ML specifications.