Open Geospatial Consortium Inc.

Date: 2007-07-02

Reference number of this OGC® document: OGC 06-010r6

Version: 1.0.0

Category: OpenGIS® Implementation Specification

Editor: Steve Havens

OpenGIS® Transducer Markup Language (TML) Implementation Specification

Document type: OpenGIS® Publicly Available Standard
Document subtype: Engineering Specification
Document stage: Final
Document language: English

(Page intentionally left blank)

Contents

Page

Figures

Page

Tables

Page

Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2, “Rules for the structure and drafting of International Standards”. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

i. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc. (OGC).

- 3eTI
- Intergraph
- Innovative Research Ideas & Services Corporation
- National Geospatial-intelligence Agency
- Oak Ridge National Laboratory
- Radiance, Inc.
- SeiCorp, Inc.
- York University

ii. Document contributor contact points

All questions regarding this document should be directed to the editor or co-editor:

CONTACT	ORGANIZATION	ADDRESS
Steve Havens (editor)	ArgonST	4220 Varsity Dr. Suite E Ann Arbor, MI 48018
Arthur Na	ArgonST	4220 Varsity Dr. Suite E Ann Arbor, MI 48018
Jon Schlueter	ArgonST	4220 Varsity Dr. Suite E Ann Arbor, MI 48018
Rachel Weingrad	ArgonST	4220 Varsity Dr. Suite E Ann Arbor, MI 48018
Valerie Yoder	ArgonST	4220 Varsity Dr. Suite E Ann Arbor, MI 48018
Don Jenkins	ArgonST	4835 University Square, Suite 15 Huntsville, AL 35816
Kurt Steele	Kurt Steele	225 E82nd, New York NY 10028
Maj. Dan Bagley	National Geospatial-intelligence Agency	Acquisition Directorate, T579, MS P-102 Airborne Projects Office 12310 Sunrise Valley Drive Reston, VA 20191-3449

CONTACT	ORGANIZATION	ADDRESS
Dr. Johnny Tolliver	Oak Ridge National Laboratory	P.O. Box 2008 M/S 6085 Oak Ridge, TN 37831
Kirk Deweese	Radiance Technologies	350 Wynn Drive, Huntsville AL 35805
Jim Greenwood	SeiCorp, Inc.	5870 Trinity Pky, Suite 350 Centerville, VA 20121
Bill Craig	SeiCorp Inc.	5870 Trinity Pky, Suite 350 Centerville, VA 20121
Mark Priest	3eTI	700 King Farm Blvd. Suite 600 Rockville, MD 20850
Gene Grindstaff	Intergraph, Inc.	P.O. Box 6695 Huntsville, AL 35824
Dr. Mike Botts	University of Alabama, Huntsville	ESSC / NSSTC Huntsville, AL 35899
Steve Liang	York University	4700 Keele St. Toronto, ON M3J1P3

iii. Additional Contributors

- Rockwell-Collins
- Air Force Research Laboratory/Sensors Directorate
- Lockheed-Martin
- US Special Operations Command (USSOCOM)
- General Dynamics
- Khoral
- Modus Operandi
- SSAI
- SAIC
- Northrop Grumman

iv. Revision history

Date	Release	Author	Section modified	Description
2003-08-25	v0.095beta	swh	all	Original TML document
2005-02-28	v.0.0.0	swh	all	Reformatted document into OGC Implementation Specification format
2006-09-21	v1.0.1	swh	all	All sections revised and updated as a result of public comment
2006-10-27	v0.0.2	swh	all	Incorporated comments of 2006-10-2 RWG
2006-11-29	v1.0.0	swh	cvr page, sec 8, Annex A	Document submitted to TC for IS vote tml namespace change
2007-07-02	v1.0.0	gsb	all	Removed all references to UTF7. Finalize formatting for Public Release

Any issues in this specification are captured in the following format

Issue Name: [Issue Name goes here. (Your Initials, Date)] **Issue Description:** [Issue Description.] **Resolution:** [Insert Resolution Details and History.] (Your Initials, Date)]

v. Changes to the OGC Abstract Specifications

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

vi. Future work

Definition of object unique id numbers (UID). TML identifies relationships among objects using UIDs. A URN scheme may be adopted for this purpose.

Definition of a physical property (phenomenon) and unit of measure dictionary

Refinement of the process object and validation test suite (Annex B)

Harmonization with other established OGC (i.e. GML, SensorML, O&M), ISO and IEEE transducer data specifications.

Other areas of future work include addressing new and expanded applications above and beyond geospatial applications; such as weather, exploration, environmental, medical, industrial, security, etc. Transducer Markup Language (TML) implementations will refine and clarify existing components within TML as well as providing more examples of how transducers and transducer systems are modeled.

Foreword

This edition of the OGC TML Implementation Specification replaces all previous (beta) TML specifications. IRIS Corporation developed TML under a government contract to enable the interoperability and fusion of dissimilar sensor data. Organizations contributing to this work are identified in section iii - Submitting Organizations, section iv - Document Contributors, and section v - Other Contributors. Section v titled “Other Contributors”, indicates organizations that contributed to the development of TML but are not OGC members.

Prior to OGC submission, the National Geospatial-Intelligence Agency (NGA) requested that the OGC investigate the adoption of TML for the exchange and archive of streaming sensor data. OGC Web Service – 3 (OWS-3) demonstrated how TML would enable the capability to discover, describe, and exchange streaming sensor data using the OGC Sensor Observation Service and the Sensor Alert Service. TML fits very well into the Sensor Web Enablement (SWE) family of services for the exchange and archive of streaming sensor data.

TML works within transducer exchange messages from multiple application domains (defense, weather, exploration, environmental, medical, industrial, security, etc.). To provide a complete picture within these messages, the TML sensor data description will be complemented with other domain specific information. For example in the defense domain, specific information would include security, bombing encyclopedia number, sortie and mission numbers, etc. However, this information would not be applicable in the medical domain.

This document includes four annexes; Annex C is informative, Annexes A and B are normative.

OGC draws attention to the fact that it is claimed that compliance with this specification may involve the use of a patent or other intellectual property right (collectively, "IPR") concerning ‘Method and apparatus for acquiring and processing transducer data’ given in the entire document. OGC takes no position concerning the evidence, validity or scope of this IPR.

The holder of this IPR has assured OGC that it is willing to license all "Necessary Claims" (as defined under the OGC IPR Policy) relating to this specification it owns [and any third party Necessary Claims it has the right to sublicense] which might be infringed by any implementation of this specification to OGC and those licensees (Members and non-Members alike) desiring to implement this specification. The statement of the holder of this IPR to such effect has been filed with OGC. Information may be obtained from:

Name of Holder of Right: IRIS Corporation (now known as ArgonST)

Address: 4220 Varity Drive, Suite E, Ann Arbor, MI, USA

Attention is also drawn to the possibility that some of the elements of this specification may be the subject of IPR other than those identified above. OGC shall not be responsible for identifying any or all such IPR.

For additional information on TML visit http://www.transducerml.org or http://www.ogcnetwork.net/node/105.

Introduction

TML defines:

- a set of models describing the response characteristics of a transducer
- an efficient method for transporting sensor data and preparing it for fusion through spatial and temporal associations

Sensor data is often an artifact of the sensor's internal processing rather than a true record of phenomena state. The effects of this processing on sensed phenomena can be characterized as functions.

TML response models are formalized XML descriptions of these known hardware behaviors. The models can be used to reverse distorting effects and return artifact values to the phenomena realm. TML provides models for a transducer's latency and integration times, noise figure, spatial and temporal geometries, frequency response, steady-state response and impulse response.

Traditional XML wraps each data element in a semantically meaningful tag. The rich semantic capability of XML is in general better suited to data exchange rather than live delivery where variable bandwidth is a factor. TML addresses the live scenario by using a terse XML envelope designed for efficient transport of live sensor data in groupings known as TML clusters. It also provides a mechanism for temporal correlation to other transducer data.

OpenGIS® Transducer Markup Language Implementation Specification

1 Scope (Informative)

This document describes TML and how it captures necessary information to both understand and process transducer data. TML is intended for communicating transducer data between a transducer node (containing one or more transducers) and a transducer processing/control device (application). Descriptions of how TML captures the actual data and descriptions of necessary information such as system calibration, transducer behavior, conditions of operation and data collection parameters critical to logical data structure model are all captured within TML. This document specifically describes the TML transducer behavior models and TML data models. It does not describe a service for delivering TML-structured data, but there is a brief discussion of SOA and streaming platform considerations.

2 Conformance (Informative)

Any TML data or software implementation claiming conformance with one of the conformance classes shall pass all test cases of the corresponding abstract test suite.

3 Normative references (Normative)

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies. In the event of a conflict between the text of this document and the references cited herein, the test of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exception has been obtained.

IEEE Std 754-1985 (Reaff 1990), *IEEE Standard for Binary Floating-Point Arithmetic.*

ISO 1000 *SI Units and Recommendations for the use of Their Multiples and of Certain Other Units*, 1992

ISO 8601:2004 *Data elements and interchange formats – Information exchange – Representation of Dates and Times*

ISO/IEC 10646:2003 *Information technology -- Universal Multiple-Octet Coded Character Set (UCS)*

ISO/IEC 646:2003 *Information technology - ISO 7-bit coded character set for information interchange*

Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation (6 October 2000)

Namespaces in XML. W3C Recommendation (14 January 1999). Available [Online]: http://www.w3.org/TR/1999/REC-xml-names-19990114

XML Schema Part 1: *Structures*. W3C Recommendation (2 May 2001). Available [Online]: http://www.w3.org/TR/xmlschema-1

XML Schema Part 2: *Datatypes*. W3C Recommendation (2 May 2001). Available [Online]: http://www.w3.org/TR/xmlschema-2/

4 Informative references (Informative)

DMA TR 8358.1 *Datums, Ellipsoids, Grids, and Grid Reference Systems*

IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*

ISO/IEC CD 18026 *Information technology – Computer Graphics and Image Processing – Spatial Reference Model*

ISO 19108 *Geographic Information – Temporal schema*

ISO CD 19130 *Geographic Information – Sensor and Data Model for Imagery and Gridded Data*

NIMA TR8350.2 *Department of Defense World Geodetic System 1984*, Third Edition 4 July 1997

OGC 03-040 *OpenGIS® Reference Manual*, 4 Mar 2003

OGC 05-086r2, *Sensor Model Language, (SensorML),* 1 Feb 2006

OGC 04-071 *Some Image Geometry Models*, 30 Sep 2004

OGC 05-008, *OpenGIS® Web Services Common Specification*

OGC Abstract Specification, Topic 2 - Spatial Referencing by Coordinates. Topic 2

5 Terms and definitions (Normative)

This section contains key terms and definitions required to understand the TML concepts.

For the purposes of this specification, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] and in OpenGIS® Abstract Specification Topic 2: *Spatial referencing by coordinates* shall apply. In addition, the following terms and definitions apply.

5.1 actuator

An actuator transforms input data into an in-situ action or phenomenon. An actuator is a subdivision of transducers.

5.2 attitude invariant transducer

An attitude invariant transducer exhibits the same response independent of its attitude or orientation. In-situ transducers, which do not have a direction associated with them, and remote transducers with an omni-directional ambiguity space are attitudinally invariant.

5.3 attribute

XML elements have attributes, which are pieces of information which appear in an XML element's opening tag. Attributes have a name and a value, and look like this:

<element attribute="value">

TML uses attributes to specify information pertinent to a specific element. Section 7 of this document lists all attributes which are associated with elements.

5.4 baseband signal

Original signal to or from a transducer prior to modulation onto a carrier or sub-carrier, or after demodulation from a carrier or sub-carrier signal.

5.5 coordinate system

A system or methodology for positioning points within a spatial reference system. TML utilizes only three coordinate systems: rectangular, cylindrical, and spherical.

5.6 cylindrical coordinate system

A 3-dimensional coordinate system with one angle coordinate (alpha) and two distance coordinates (r, z), where r is the length of the line between the normal projection of the point onto the xy plane and the origin. The two dimensional version of the cylindrical coordinate will be referred to as a polar coordinate system utilizing only the α and r coordinate.

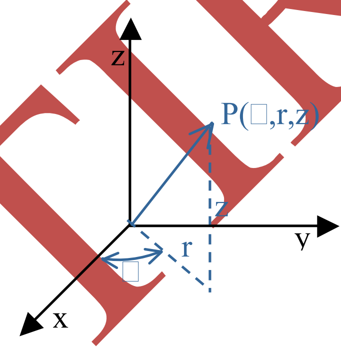


Figure 1 - Cylindrical coordinate system

5.7 data description/metadata

The data description/metadata is a complete picture of the data and how to process the data. It is comprised of all information contained in the system, transducer, relations, process and data cluster tags. The data description is a self-contained description of the TML data clusters with information about the transducer, its behavior, information pertaining to the conditions of operation which might be relevant to later processing of data, and the layout of the data within the TML data stream.

5.8 data index

A count to uniquely identify each data set (e.g. pixel) location with a unique index or count within a characteristic frame (e.g. image frame). The index number is derived from the logical serialization order of the datasets within the TCF data Array. The logicalStructure element within the tcfDataArray element describes the logical sequencing order in which datasets are indexed within the TCF Array.

5.9 data product

Data products refer to images, reports, analysis and other information produced from processing raw transducer data. Data products may be in formats other than TML and the TML description can support describing other transducer formats. However, the accuracy and understandability of the data can not be assured in these cases.

5.10 dimensionality

Data captured in a transducer characteristic frame has n-dimensions (or dimensionality). The dimensionality give clues to the recipient processor how multi-dimensional data may be represented. The dimensionality may also help prepare for processor memory allocation. The characteristic frame logical data structure is described using an array, set, and units data constructs. A zero dimensional structure will contain one or more units in a set. With only one set in the only existing array. A one dimensional structure will contain one or more data units in a set with two or more sets in a single array. Two dimensional data has an array of two or more subordinate arrays each containing two or more sets. Similarly, a three dimensional structure is the extension of a two dimensional structure. Time is not a dimension.

5.11 earth centered earth fixed (ECEF) spatial reference system

The origin of this reference system is located at the center of mass of the earth. The right hand orthogonal axis x, y, and z are oriented with x at the prime meridian and z north. Objects in the ECEF spatial reference system may be positioned using either the rectangular, spherical, or the WGS-84 geodetic coordinate system

NOTE: Reference WGS-84

5.12 ellipsoidal height geodetic /height /altitude/ elevation

The elevation or altitude is the distance of a point from the surface of the reference ellipsoid along the direction of a vector normal to the ellipsoid surface.

NOTE 1: Ellipsoidal height is defined with respect to. an ellipsoidal model of the shape of the earth. Ellipsoidal height is measured from the point along the line perpendicular to the ellipsoid's surface. Gravity-related height is defined with respect to. the geoid model of gravity. Gravity-related height is measured from the point along the line perpendicular to the geoid surface.

NOTE 2: Height or altitude of a point outside the surface is treated as positive; negative height is also named as depth. This is an exception that needs to be noted because the local reference system xyz axes are aligned with north (x), east (y), and down (z).

5.13 Euler angles

Three rotation angles (ω,φ,κ) describing how a child spatial reference system is oriented relative to a parent spatial reference system. When describing the child orientation relative to the parent the angles are derived in the following order, κ,φ, ω (i.e. heading, pitch, and roll). When describing the parent orientation relative to the child the angles are derived in the following order, ω,φ,κ (i.e. roll, pitch, and heading).

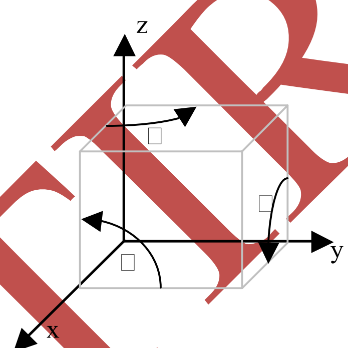


Figure 2 - Euler angles

5.14 geocentric coordinate reference system

A spherical reference system based using the ECEF spatial reference system. The coordinates used to specify a point are: geocentric latitude, longitude and radius. The radius is the distance from the origin of the ECEF reference frame to the point. Longitude is the same as the alpha angle in the spherical system, latitude is the same as the beta angle in the spherical system, and radius is the same as rho in the spherical system.

5.15 geocentric latitude

The geocentric latitude coordinate is the angle between a line defined by a point on the earth surface and the center of the ECEF reference system and the equatorial plane. The geocentric latitude is the same as the beta angle within the ECEF reference system.

5.16 geodetic coordinate reference system/ellipsoidal coordinate system

A method for describing the three dimensional position of a point within the ECEF spatial reference system. The coordinates used are geodetic latitude, longitude, and geodetic altitude. The coordinates are defined using elliptical coordinates; see the definition for geodetic latitude, longitude, and geodetic altitude).

5.17 geodetic latitude/ latitude/ ellipsoidal latitude

The geodetic latitude coordinate is the angle between the normal (line perpendicular to) to the earth reference ellipsoidal surface and the equatorial plane.

5.18 Gregorian calendar

Calendar in general use first introduced in 1582 to correct an error in the Julian calendar.

NOTE: In the Gregorian calendar, common years have 365 days and leap years 366 days divided into 12 sequential months and is designed to keep synchronization with the tropical year.

5.19 latitude/geodetic latitude/ellipsoidal latitude

Angle from the equatorial plane to the perpendicular to the ellipsoid through a given point, with northwards as positive.

5.20 linear time invariant system/ LTI system

A linear time invariant transducer or process is both time invariant and linear. A transducer or a process is time invariant if a time shift in the input results in the same time shift of the output. A transducer or a process is linear if it exhibits the criteria for superposition. Superposition requires that the relation between the input and the output have two properties, additivity and homogeneity. Additivity implies that if an input *a* results in output *a′* and input *b* results in output *b′,* then input *a*+*b* results in output *a′+b′*. Homogeneity implies that if input *a* results in *a′*, then input *k(a)* results in *k(a′)*, where *k* is a constant.

5.21 linearity

A property of the steady state transfer function. The linearity of a transducer is described in terms of percentage of how far the response deviates from a perfect linear response over the operating range of the transducer.

5.22 live/ historical (antonym)

This term when used in association with transducer data implies that something is happening at the same time or very close to the same time as original real world phenomena or events. It has to do with the latency time between real world events and using the data. Data can be processed or viewed live or you can process or view archive data. Viewing transducer data is considered live if the display were an accurate representation of the current events of phenomena. If the playback or viewing of transducer data represents data from a past time, then it is considered historical data.

5.23 local earth spatial reference system

The origin of this reference system is located on the surface of the earth (WGS-84 ellipsoid model). If the coordinates are to be references in other local references or relative to ECEF reference, then the local earth spatial reference system must be qualified with a specified geodetic latitude and longitude coordinate specifying the origin. The right hand orthogonal coordinate axis x, y, and z are oriented in the north, east, and down direction respectively. Objects in the local earth spatial reference system can be positioned using either the rectangular or spherical coordinate system

5.24 location invariant transducer

A location invariant transducer will exhibit the same response independent of its location.

EXAMPLE: Sensors which measure their attitude relative to a spatial reference system are location invariant. The location of that sensor is unimportant, only the attitudinal relationships are important.

5.25 Longitude/geodetic longitude/geocentric longitude/ellipsoidal longitude

Angle from the prime meridian plane to the meridian plane of the given point, with eastward treated as positive.

NOTE: The geodetic longitude is the same as the geocentric longitude which is the same as the α coordinate in the ECEF Reference System using the spherical coordinate system.

5.26 modulation

Modulation is the process of putting a lower frequency in a higher frequency signal. There are several processes for modulation such as amplitude, frequency, phase, and variants thereof.

5.27 phenomenon/physical phenomenon property

A phenomenon is a physical property that can be observed and explained by physics. Phenomenon relate to properties of matter, energy, and space-time. A phenomenon can be observed by a sensor or changed with an actuator. A phenomenon may be spatially and/or temporally modulated. In other words, its value may change as a function of location and time [f(x,y,z,t)]. Images are created by representing the spatial modulation. Phenomenon can be artificially created. The phenomenon generated [f(g(x,y,z,t))].by artificial means may be a function of the input data to a transmitter or actuator [g(x,y,z,t)].

As part of understanding what transducer data is, the proper understanding of the phenomena and phenomena properties is critical. The following list is only a small sample of the phenomena and properties commonly used. OGC shall maintain a library of phenomena and phenomena properties.

5.28 positional invariant transducers

A positional invariant transducer may be either locationally invariant or attitudinally invariant. A non-positional invariant transducer is neither locationally or attitudinally invariant. If a transducer is locationally invariant then the location of the transducer is unimportant. If a transducer is attitudinally invariant then the attitude of the transducer is unimportant.

EXAMPLE 1: If an in-situ temperature sensor measures the same temperature independent of what orientation it is in then this transducer is attitudinally invariant. This would imply that there is no shaped field (i.e. ambiguity space) around the transducer. It must be either omni-directional or truly a point (in-situ) detector.

EXAMPLE 2: If we know transducer 1 is oriented in a particular orientation relative to transducer 2, and positioned relative to transducer 3, and transducer 2 measures its own orientation relative to an absolute datum, then to determine the attitude of transducer 1 the position of transducer 2 is unimportant. The fact that the orientation of transducer 1 is fixed relative to transducer 2 implies a ridged body and every point on that rigid body experiences the same angular rotations, therefore the position of the attitudinal measurement form transducer 2 is unimportant.

5.29 real-time, less than real-time, greater than real-time

This term refers to the playback or viewing rate of transducer data. For example if data took one minute to collect and the viewing or playback time took one minute, then this would be considered real-time. It is independent of live or archive data. Obviously viewing live data is typically real-time unless the viewing is slowed down, then it becomes archive data after a short time. If data viewing is slowed then it becomes less than real-time. If data viewing is sped up then it becomes greater than real-time. A distinction should be drawn between real-time and live; real-time refers to the rate of data playback while live refers to data playback at the same time that it is captured.

5.30 rectangular coordinate system, Cartesian coordinate system

A set of orthogonal coordinates specifying the location of a point relative to a spatial reference point (origin). The coordinates are specified by normal projections of the point onto the three cardinal axes (x, y, & z). The displacements of these projections represent the three rectangular coordinates for specifying the three dimensional location of the point. The rectangular coordinates are represented by specifying the linear displacement of the intersections of the normal projections from the point and the three cardinal axes.

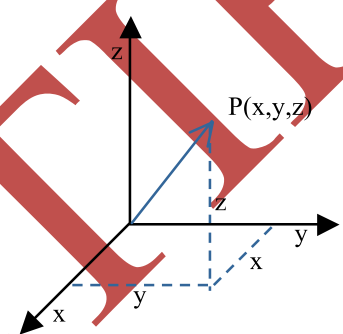


Figure 3 - Rectangular coordinate system

5.31 reference ellipsoid

An ellipsoid used as the best local or global approximation of the surface of the geoid. The reference ellipsoid is the reference surface for calculating the geodetic latitude and geodetic height.

5.32 saturation

Saturation is the exact point of the response model of an LTI system where increasing the phenomenon no longer corresponds to an increase in the data value

5.33 sensitivity

Sensitivity is the minimum level of a phenomenon's property to which data can correspond.

5.34 sensor receiver

A sensor receiver is a specific type of transducer for detecting and measuring a phenomenon. A transducer which converts a physical phenomenon into a digital data representation.

5.35 spatial coordinate reference system

A spatial system containing both a spatial reference system and a coordinate reference (datum) system

5.36 spatial coordinate system, coordinate system

The methods by which a three dimensional point is can be specified within a spatial reference system. The various coordinate systems are: rectangular coordinate system (see definition), spherical coordinate system (see definition), geodetic coordinate system (see definition), and the cylindrical coordinate system (see definition).

5.37 spatial invariance

Spatial invariance relates to internal geometry models. An internal geometry model must be spatially invariant, meaning the spatial characteristics of the model are independent of the location or attitude of the transducer anywhere in the world.

5.38 spatial reference system, spatial datum

This specifies the origin and attitude of the right hand orthogonal spatial coordinate axis frame. Different spatial reference (datum) systems are:

- ECEF spatial reference system (see definition)
- earth local spatial reference system (see definition)
- transducer spatial reference system (see definition)

5.39 spherical coordinate system

1) the first angular coordinate alpha (α) is derived by the angle of rotation of the xz plane about the z axis until the plane contains the point. The second angle beta (β) is derived by the angle of rotation of the rotated xy plane about the y axis until the plane contains the point. The third length coordinate rho (ρ) is the distance from the point to the origin along a line containing both the point and the origin. 2) a spatial coordinate specified by two (2) angle coordinates and one (1) length coordinate relative to a spatial reference system.

NOTE: Not to be confused with an ellipsoidal coordinate system based on an ellipsoid 'degenerated' into a sphere.

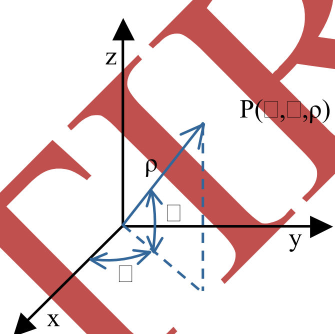


Figure 4 - Spherical Coordinate System

5.40 standard units of measure, units of measure UOM

The defined quantity in which dimensioned parameters are expressed. Such as Kilograms, meters and seconds. OGC shall maintain a dictionary of Units of measure which shall be coordinated with the Phenomenology dictionary

5.41 streaming data

Streaming data is data which has a time component. Data within the stream may occur at different instances in time. The TML data stream time stamps each data cluster so that the data can be utilized in the proper time sequence. The data may be live or archive. If the data is live, the end of the file is not yet known. Applications should not wait until the end of file to utilize the live data. If the data is from an archive then obviously the file length is known.

5.42 temporal coordinate reference system

A temporal measurement system which contains references to both a temporal reference (datum) and a temporal coordinate systems (coordinate units).

5.43 temporal coordinate system

A set of one or more coordinates specifying the time of an event relative to a temporal reference (datum) system. Common temporal coordinate systems include the Gregorian calendar, with: years, months, days, hours, minutes, and seconds.

5.44 temporal invariance

Relating to internal geometry models. An internal geometry model must be temporally invariant. This means the internal temporal characteristics of the model are independent of the time and date. The temporal characteristics are defined relative to an internal temporal reference system such that the temporal characteristics are temporally invariant to external temporal reference systems. The temporal characteristics look the same on Monday as they do on Tuesday. They appear identical regardless of time.

5.45 temporal reference system, temporal datum

The reference event to which an event is related to in time. Either the reference can be an enterprise absolute temporal reference, such as the 1970 Epoch, or it may be a relative reference such as the time of a transmitter trigger pulse, to which received data is time referenced. Common reference systems are the Gregorian calendar reference in which we are in the year 2005, UTC time reference in Greenwich UK, and the Epoch on 1 Jan 1970.

5.46 TML Transducer Mark-up Language

A transducer exchange language described in this specification

5.47 TML application (processor/controller)

A TML processor is a computing machine to understand and process the TML message, a TML controller is a computing machine to generate TML messages. A single computing device may contain one or both functions. (See **Figure 7** - Generic TML implementation) A processor/controller can also be a process node on a network - providing services to clients on the network.

5.48 TML application node

A node presents TML data on a network for TML applications. It may use a set of common software libraries that enable a user application to ingest, process, display, and create (control) TML data. (See **Figure 7** - Generic TML implementation)

5.49 TML network node

A network node for the distribution of live and/or historical TML data. The network node may federate multiple network, transducer, and process nodes. (See **Figure 7** - Generic TML implementation)

5.50 TML process node

A process node presents TML processes on a network for process systems. This node may contain adapters for transducers, processes, system clock, network, archive, and registry. The transducer node will manage and maintain data description for the data it is handling, and system clock and timing. (See **Figure 7** - Generic TML implementation) The process node will input TML data and output processed TML data.

5.51 TML transducer node

A node which presents transducer information on a network for transducer systems. This node may contain adapters for transducers, processes, system clock, network, archive, and registry. The transducer node will manages and maintain data description for the data it is handling, and system clock and timing. (See **Figure 7** - Generic TML implementation)

5.52 transducer adapter

Transducer adapter is an adapter that provides the interfaces between a transducer and the TML transducer node. A unique adapter will need to be built for each unique transducer interface. This puts any proprietary or unique interface behind the TML transducer node such that the proprietary or uniqueness of the interface is transparent to the rest of the network. The TML adapter will contain the TML data description if it does not come from the transducer. The adapter will contain the timing and synchronization processing to ensure the proper time relations exist between the data and the phenomenon. For example, the time stamped on sensor data needs to represent the time of the phenomenon instance and not the time the data is outputted. For transmitters and actuators the application of the phenomenon instance needs to occur at the precise time of the indicated time tag or as soon as possible in the absence of a time tag. The adapter will group transducer data into data clusters for receivers or sensors. IEEE P1451 Smart transducer interface will greatly simplify this task and make the number of adapters required to be developed much less. Only one adapter will need to be built for all IEEE 1451 transducers. (See **Figure 7** - Generic TML implementation)

5.53 transducer data event, transducer event

An event (usually the result of an analog to digital conversion trigger) which samples the state of a sensing transducer or an event that commands an actuator. A trigger source can provide periodic triggers or it can provide a trigger only when certain criteria are met such as a threshold condition. An event will have a location and a time.

5.54 transmitter

A subdivision of transducers. A transmitter transforms input data into a remote action or phenomenon. Normally thought of in conjunction with radio electro-magnetic energy, but can apply to other propagating energy forms (e.g. acoustic).

5.55 UTC - Coordinated Universal Time

Time scale maintained by the Bureau International des Poids et Mesures (International Bureau of Weights and Measures) and the International Earth Rotation Service (IERS) that forms the basis of a coordinated dissemination of standard frequencies and time signals.

5.56 WGS-84 geodetic coordinate system

Coordinate system in which position is specified by geodetic latitude, geodetic longitude and (in the three-dimensional case) ellipsoidal height, associated with the ECEF spatial reference system. The geodetic latitude is the angle between the equatorial plane and a line perpendicular to the ellipsoid surface in the plane of a constant meridian longitude. The origin and the ellipsoidal parameters (semi-major and semi-minor axis) are specified in the WGS-84 documentation.

NOTE 1: See ellipsoidal or geodetic coordinate system in OGC Abstract specification Topic 2.

NOTE 2: See geodetic latitude. geodetic latitude ≠ geocentric latitude (spherical β angle).

Issue Name: tml rwg 1.13 (S. Cox, 3 Oct 2006)

Issue Description: Some of these definitions appear in the definitions maintained by ISO 19104. The local dictionary should not contain definitions different from these.

Resolution: Identify as an issue and document as future harmonization work with ISO specifications (S. Havens, 3 Oct 2006)

6 Conventions (Informative)

6.1 Normative Verbs

In the sections labeled as normative within this document, the key words "must", "must not", "required", "shall", "shall not", "should", "should not", "recommended", "may", and "optional" are to be interpreted as described in Internet RFC 2119 [1].

6.2 Abbreviated terms

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 05-008] apply to this document, plus the following abbreviated terms.

Table 1 - Abbreviated Terms

Abbreviation	Term
A/D	Analog to Digital
API	Application Programming Interface
CCD	Charge Coupled Device
COTS	Commercial Off The Shelf
CF	Characteristic Frame
CGI	Common Gateway Interface
CRS	Coordinate Reference System
CS	Coordinate System

DCP	Distributed Computing Platform
DTD	Document Type Definition
DTED	Digital Terrain Elevation Data
DTM	Digital Terrain Model
ECEF	Earth Centered Earth Fixed
EPSG	European Petroleum Survey Group
FTP	File Transfer Protocol
GIF	Graphics Interchange Format
GIS	Geographic Information System
GML	Geography Markup Language
GPS	Global Positioning Service
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
IFOI	Instantaneous Field of Influence
ISO	International Organization for Standardization
IETF	Internet Engineering Task Force
IMU	Inertial Measurement Unit
JPEG	Joint Photographic Experts Group
LDS	Logical Data Structure
LTI	Linear Time Invariant
MIME	Multipurpose Internet Mail Extensions
OGC	Open GIS Consortium
OWS	OGC Web Service
PCF	Process Characteristic Frame
PDA	Personal Digital Assistant
PSD	Power Spectral Density
RPC	Rational Polynomial Coefficient
RSM	Replacement Sensor Model
SPS	Sensor Planning Service
SSL	Secure Socket Layer
SWE	Sensor Web Enablement
TCF	Transducer Characteristic Frame
TCP/IP	Transmission Control Protocol
TIFF	Tagged Image File Format
TSM	Tactical Sensor Models
UDP/IP	User Datagram Protocol/Internet Protocol
UID	Unique IDentifier
UML	Unified Modeling Language

URL	Uniform Resource Locator
UTC	Universal Time Coordinates
WGS	World Geodetic System 84
XML	Extensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol

6.3 Platform-neutral and platform-specific specifications

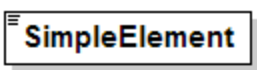
The TML specification is platform neutral. There are no platform specific requirements.

6.4 XML schema notation

Most diagrams that appear in this specification are presented using an XML schema notation defined by the XMLSpy[1] product and described in this subclause. XML schema diagrams are for informative use only.

6.4.1. Element

A named rectangle representing the most basic part of the XML Schema notation. Each represents an XML "Element" token. Each Element symbol can be elaborated with extra information as shown in the examples below.



This is a mandatory simple element. Note the upper left corner of the rectangle indicates that data is contained in this element.

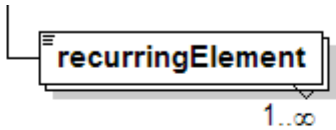
6.4.2. Optional Element

Optional (non mandatory) elements are specified with dashed lines used to frame the rectangle.

[1] XML Spy: http://www.altova.com

6.4.3. Recurring Element

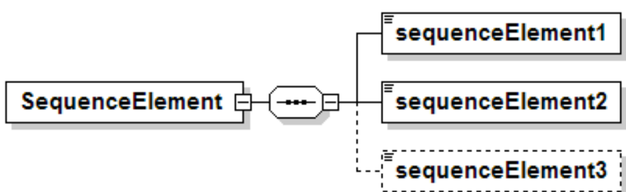
This element (and its child elements if it has any) can occur multiple times.



This example shows a recurring element that must occur at least once but can occur an unlimited amount of times. The upper bound here is shown with the infinity symbol.

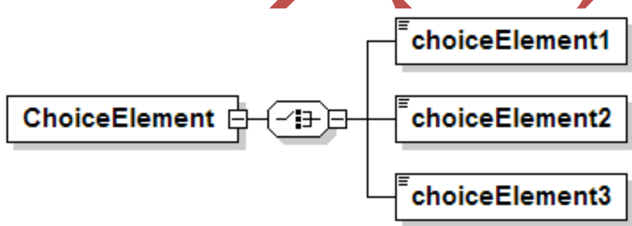
6.4.4. Sequence Connector

The connection box, called a sequence indicator, indicates that the “SequenceElement” data is made up of three elements. In this example, the first two elements are mandatory and the third element is optional



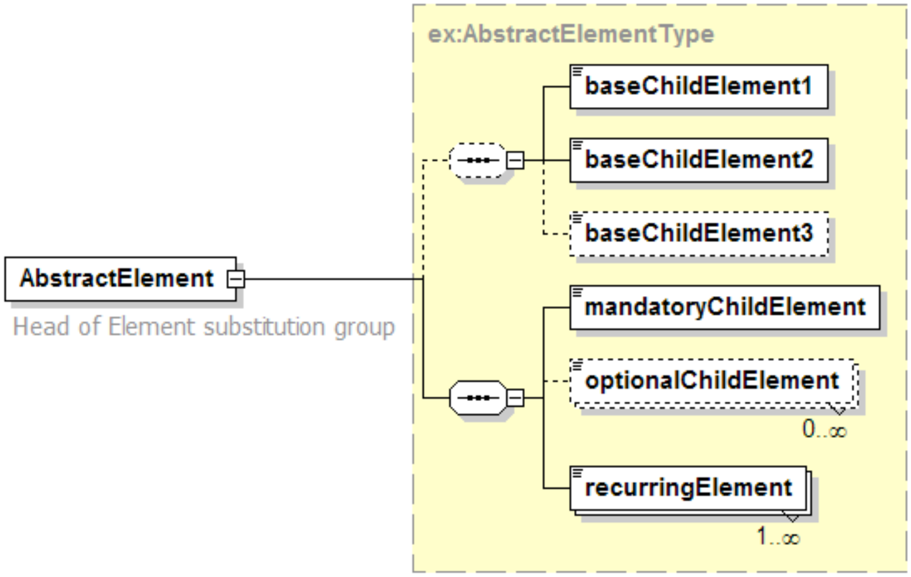
6.4.5. Choice Connector

The connection box here is a “choice” indicator, indicating that there is always going to be exactly one of the child elements listed on the right.



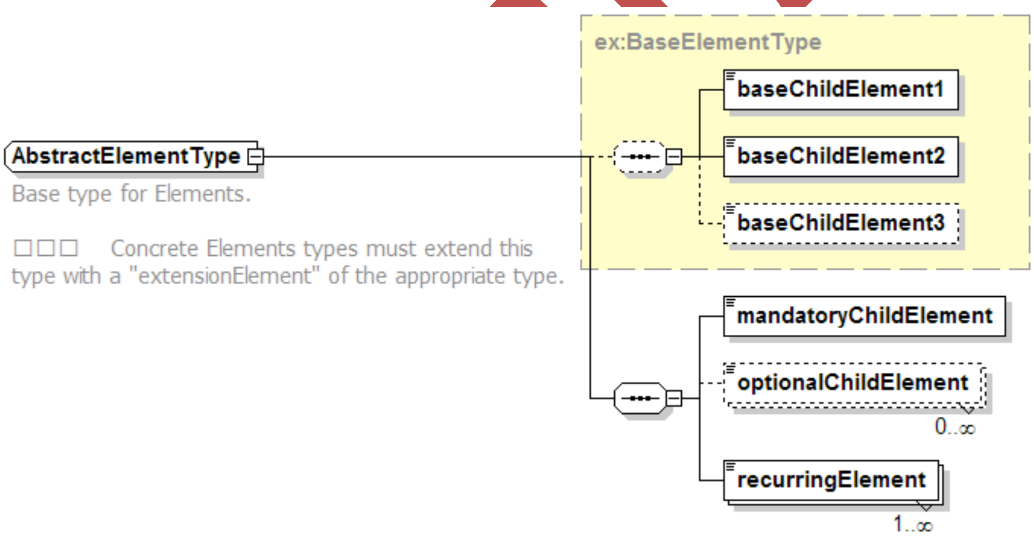
6.4.6. Definition with Complex Type

This diagram illustrates the use of a complex type (i.e., “ex:AbstractElementType”) for defining an XML element (e.g., “AbstractElement”).



6.4.7. Complex Type

This diagram illustrates the definition of a complex type (i.e., “AbstractElementType”), extending another complex type (i.e.,“ex:BaseElementType”) with three additional elements. Complex types can be reused to specify that different elements are of the same type.



7 TML Concepts (Informative)

7.1 TML Introduction

Transducer Markup Language (TML) is a language for capturing and characterizing not only data from transducers, but information necessary for the processing and understanding of that data by the eventual recipient of the transducer data. Both sensors and transmitters can be captured and characterized within TML, leading to the use of the term “transducer” rather than “sensor”. TML handles not only static but also streaming transducer data. TML permits the data stream to handle live transducer data both being added to the stream and being deleted from the stream.

Descriptions relevant to the later processing and understanding of the data, including calibration, operational conditions, device settings, transducer properties and characteristics, relationships of transducers to each other in a multi-component system and system behavior are all among the properties captured in a TML description. Logical models, behavioral models, transfer functions, and other information critical to the processing of data are all captured within TML.

TML is capable of precise time-tagging of data, so that it is possible to know precisely when a physical phenomenon was measured at the individual measurement level, and also captures latency or delay information at a fine resolution. This enables the precise determination of when a data point was taken, as well as aiding in interpolation between data points and the reconstruction of events.

7.2 Interoperability and Integration of Transducer Data

The primary purpose of TML is to enable widespread sharing while providing greater understanding of transducer data, such that we can gain a better picture of our world through transducers. Transducers are our interface to the real world. Just as our minds rely on multiple senses to gain a better situational understanding of our environment so must we utilize multiple modalities of transducers to gain a better situational understanding of our world.

To manage the complex task of processing and associating millions of sensory inputs, we must utilize computing machines to coordinate and associate data and represent only interesting data in a summarized manner such that humans can make higher-level decisions in a timely manner. To facilitate this task we must enable a seamless and unambiguous communication language for transducers and computing machines to act as a homogeneous system. To accomplish this a transducer communication language must be complete, consistent, and efficient as possible. Many times these properties are mutually exclusive.

TML is an interoperable solution for the exchange, archive, and common understanding of transducer data

TML is scalable to facilitate a wide variability of application.

7.3 Interoperability and Common understanding of Transducer Data

For interoperability we need:

- the data which describes the encoding, structure and organization of the transmitted data
- the data that describes the structure and organization of the transducer data

For transducer data understanding we need:

- the data which describes the object of the measurement or action of the transducer is
- the data which describes the transducer response and geometric characteristics
- the data which describes any pre processing to the data
- the data that describes external relationships for the transducers and processes within a system

TML provides a cohesive set of common normalized descriptions to enable the common understanding of transducer data. This common understanding will enable machines to begin to make lower level associations of transducer data (such as conflation or association and registration of features). It is paramount that the data description documents remain closely linked with the data, one cannot be understood without the other.

TML enables the description to provide context to transducer data. Transducer data relates to a physical property (or phenomenon) which relates to a physical subject. The subject of the property that the transducer interfaces may or may not be known at the time the TML data descriptions are written. If they are not then they can be described later in time. Data to and from transducer may be pre processed. Dealing with data which will be processed or which has been processed requires a complete description of the processing. These relationships are shown in Figure 5. A transducer system may be composed on one or more objects. An object can be a transducer system itself, or a transducer, process, or subject. This relationship is also shown in Figure 5.

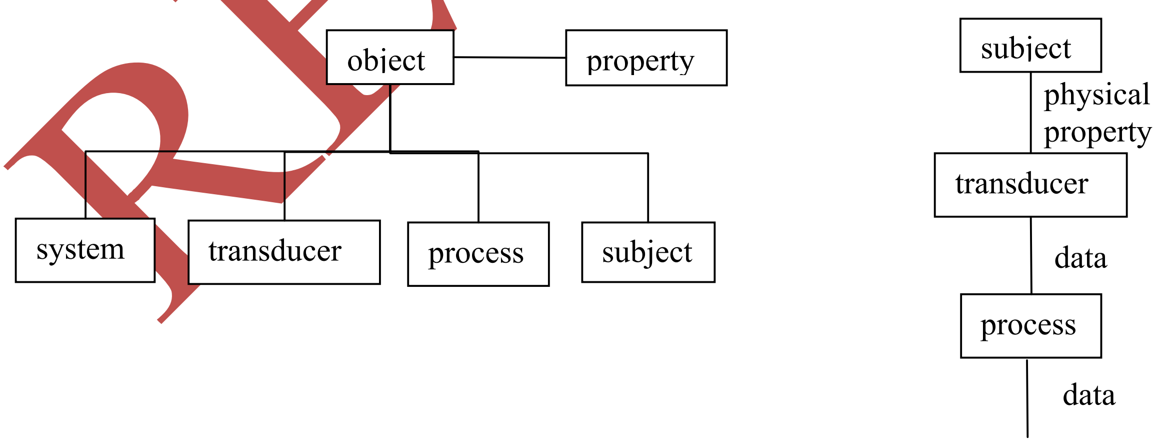


Figure 5 - TML Conceptual Components

7.4 Live and Historical Streaming Data

Because transducers are real world interfaces, their responses are representative of the changing world. This changing world is represented in live streaming transducer data. TML captures this living streaming data, which is representative of real world transducer events corresponding to multiple phenomena and maintains the relative and absolute temporal and spatial relationships of the data such that we can review the events exactly as they happened either live in real time or at a different place and/or a different time. TML data represents a continuous stream of data from or for possibly a multitude of different transducers, all interleaved randomly, in roughly chronological order. It is up to the data acquisition equipment to capture the data as precisely and accurately as possible such that the precision and accuracy capabilities of the transducers are fully recognized and the acquisition process is not degrading the data in any fashion. The task of data integration and understanding is left to the transducer processors.

7.5 Variability of Transducers from Simple to Complex

There is a great variability in the complexity of transducers and the applications that process and control them. A transducer system may be composed of several different transducers as well as processes that prepare the data. Transducers as simple as a temperature probe and as complex as an interferometric synthetic aperture radar system must be integrated into a single homogeneous transducer representation of our world. Transducer work in two directions, data not only comes from transducers (e.g. sensors) but also can be sent to a transducer to produce certain predetermined results (e.g. hydraulic actuator).

7.6 Variability in uses and applications

Transducers have widespread use in medical, mineral exploration, environmental, industrial process control, military reconnaissance and surveillance, building security, weather, and air space control, to name a few. There is also great variability in complexity of processing transducer data. TML supports complex as well as simple processing and it is domain or application agnostic. In other words, it does not make any assumption on how the data will be processed, displayed, or used, nor does it contain any domain specific administrative data. TML is ideally suited for sharing transducer data across application domains. Applications may also demand that transducer data be from live sources as well as from archived sources. Live sources of transducer data may be from mobile, portable, or stationary transducers. TML handles all the variability in uses and applications efficiently.

Figure 6 below displays how TML could work in exchanging transducer data across application domains. The data associated with the transducers can be consistent across domains. The "administrative" data associated with the transducers may be different across domains. The administrative data is data for using the transducer data more efficiently within an application domain such as national defence. Is this domain examples of administrative data would be Mission number, Wing, squadron, essential elements of information, requestor of information, etc. In the medical domain administrative data such as patient number and doctor or examiners name may be examples..

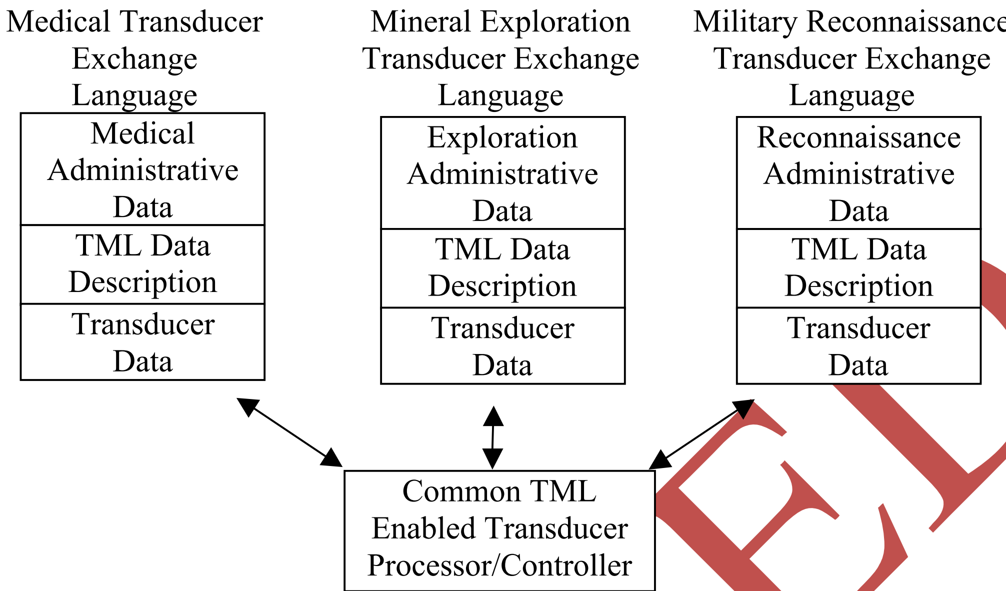


Figure 6 - Sharing of transducer data across application domains

Other than a general philosophy of 'plug and play' and 'composable' systems, TML is designed to be independent of object, pattern or process design methodologies.

TML transports arbitrary data formats from a sensor platform because it is not always practical to adopt a standardized data encoding. The following illustrate some cases where this is of use:

- Many sensor manufacturers already have arrangements with third party software firms to design the output format.
- Changing the output format of a sensor may impact existing deployments.
- Sensors often come with software packages meant to act as the client itself, or as the interpreter of the data format.
- Sensor data may have security restrictions requiring a private key mechanism for decoding. In this case the format cannot be revealed for purposes of developing application schemas.

It is paramount to preserve raw transducer data in as close a manner to the original form as possible. Sensor data is often an artifact of the sensor's internal processing rather than a true record of phenomena state. For example, radar systems frequently capture a magnitude image but all of the phase information from the received waveform is absent from the magnitude image. Some applications require only the radar magnitude image while others require both magnitude and phase data. Situations such as these make it prudent to retain original unaltered data whenever possible prior to any processing.

Other situations may require preservation of original data as well as modification of the original output or the interpolation between successive transducer measurements to determine a value at a particular time. TML's response models allow for the capture of original data and the characterization of data as close to the original form as possible.

TML is designed for delivery of data from live sensor platforms. In the sensor-to-client communications path, TML generally precedes translation to richer semantic encodings such as GML or Observations and Measurements. The latter encodings are more geared toward data exchange or scenarios in which Filter Encoding queries may be desired.

As new taxonomies are developed in the geospatial arena the need for domain agnostic representations like TML becomes greater.

7.7 Variability in communications

TML complies with the open system philosophy of protocols. TML may be carried by any session level protocol on a network (e.g. FTP, HTTP, XMPP, etc.). This also means that TML may be used with any network or transport protocol such as UDP/IP or TCP/IP. TML is scalable such that it works for passing transducer data with the simplest physical connections such as RS-232 over a point-to-point connection. TML does not provide access control, or error detection and correction.

7.8 Variability in Processing

Within a global transducer web architecture there is the potential for a great variability in the transducers, transducer data, applications, as well as processing capabilities. Processing platforms can range from cell phones and PDA to powerful multi-processor workstations and mainframe computers operating in both central and distributed processing architectures. The less capable PDA will not be able to handle the highly complex data from sophisticated transducers. There must be a procedure for handling this situation. By matching the complexity of the data with the processing capabilities of the processing machine, the user will know immediately if his machine is capable of handling the data or if they should look for some processing help on the network. TML implements a concept of defining a *complexity level* for transducer data. This *complexity level* is based on the computing resources required to process the data in *real-time*. The complexity level is based on communication resources required for communication the data. See section 7.7 variability in communications.

7.9 Static and Dynamic Data

TML uses a programmable data structure that adapts to each individual transducer. It also uses a common model or description to describe the various complexities of the transducer data. A primary objective of TML is to exchange the transducer data as accurately, and efficiently as possible. This is accomplished by exchanging two types of data

- the dynamic data which represents the time varying world

- a description of the data such that a TML enabled processor would be able to understand the various complexities of the data for a particular transducer system

The data description is required to understand the data and the data is required to understand the data descriptions. Because of the very common possibility that data description data is changing it relies on the streaming data to carry the time sensitive parameters as sensor data. It is imperative that a logical connection is kept between these two types of data.

7.10 Normalization of Phenomena

It is paramount that a normalized database of phenomenon and phenomenon properties be established for the critical task of feature recognition from multiple modalities of sensing. One key to this fusion of data and understanding is a well thought out phenomenology database.

7.11 TML Generic Architecture Components (a systems view)

This section is informative. Its purpose is to give the reader an example implementation so they may understand how TML may be used in an operational architecture. Figure 7 - Generic TML implementation is an example high-level system view of a generic architecture showing the various components. TML data can be exchanged directly (point-to-point) between a *Transducer System* and a *Transducer Processor/Controller*. If the TML transducer node has a network adapter, then TML data may also be routed through a network. TML *transducer nodes* may support services such as registry searches of local archives through persistent or stateless network connections. Figure 7 - Generic TML implementation shows an example implementation with four different adapters to the TML data. The adapters are the transducer, process, network, and application adapters. All of these adapters may utilize many of the same functions. The application adapter will interface the TML stream to the client application, which may be a transducer processor, controller, or both. To the user, data from the *transducer system* is received in TML and the transducer interface is transparent to the application. The unique interfaces to the transducers and processes are adapted to a TML interface through a TML transducer and process adapter, respectively. TML Network Nodes provide facilities for managing multiple process and transducer systems on the network. They will enable a client user to choose which transducers and processes they need to accomplish their required task. All *transducer systems* and *process systems* connected to the network should be available to any user on the network (pending security constraints) because TML provides a common interface. All of the proprietary and unique sensor interfaces are hidden from the user. That difficult task was assigned to the *Transducer System* integrator to ensure the proper *Transducer Adapters* were installed to understand the unique communication with each *transducer* in the *system*.

The TML Application Nodes are a common set of software libraries that form the interface to the TML data. These libraries will support parsing, decoding, workflow chaining and basic TML processing and control and can be used by a wide variety of applications. It will be necessary for TML data to be processed and controlled by a variety of applications. For each application an application adapter will be required such that the application can utilize the functions provided by the TML Application Node.

To take advantage of TML universality there will need to be services on a network that will enable search and discovery of transducer capabilities, transducer system live data feeds, transducer archive data, transducer processing services, security, transducer system planning, and scheduling services.

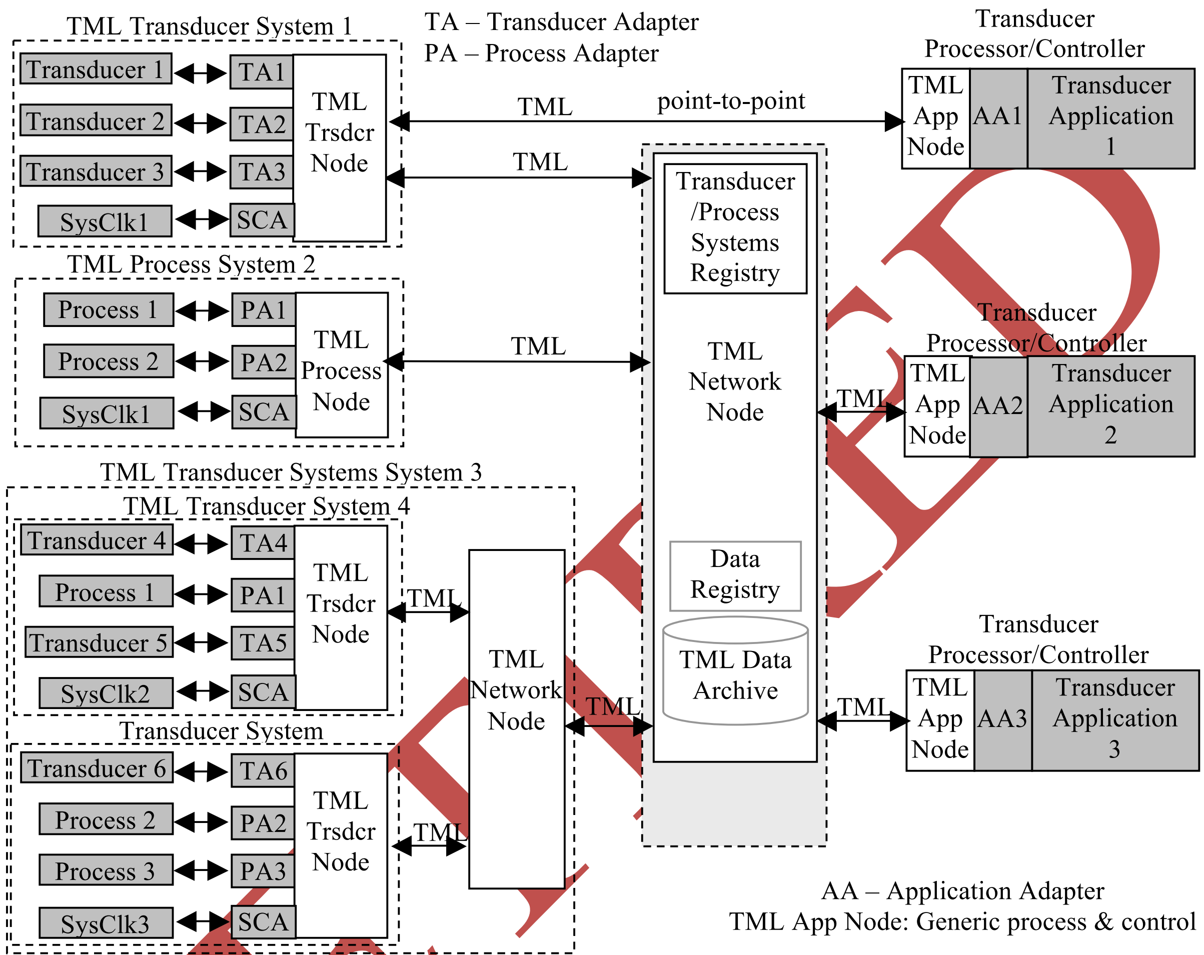


Figure 7 - Generic TML implementation

Fundamental to any distributed network architecture is the ability to find and uniquely locate resources on the network. A transducer network is no exception. Web clients must be able to search on a system registry, transducer registry, and process registry.

7.12 Modular descriptions

TML modularizes the data documentation such that each stakeholder can encapsulate documents without modification. Many people may be involved in the development of the system description documentation and not all these people will develop documentation at the same time. Transducer manufactures will develop a transducer description for their transducer. Transducer processing services will develop documents to describe their processes, and system integrators will develop documents to describe their systems. When building a system one must be able to include the descriptions, without modification, created by the individuals who designed or integrated the sub components that are being put into a system. A system integrator will then add to those descriptions of the subcomponents to the larger system. For this reason, the system integrator must be able to reuse characterizations done by other people and insert them into the descriptions when using their components.

7.13 TML data stream

The TML data stream is a product generated from a TML system. The TML data stream carries the time varying data representing various external and internal phenomena. The opening TML tag initiates a stream. A stream may be promptly terminated at any point at which time the reading machine should add a closing TML tag to make the terminated stream valid XML. If the stream is terminated normally a closing tag from the sender will terminate the stream.

TML implements a time tagged implementation of XML. What this means is that a system time clock count is inserted into the start tag of TML data elements to signify the relative time (from the sysClk) (or absolute time, in some cases) to indicate when the data contained in each TML element was acquired. The system clock should be of sufficient resolution to adequately relate time differences at a transducer sample sub-sampling interval (approximately an order of magnitude faster that the fastest sample clock in the system) and enough digits to minimize the possibility of a roll over.

In the example system below, Figure 8, five transducers (Camera, Image Size, GPS, IMU, & Compass) and two processes (JPEG Compress and Base64 encode) generate five types of data clusters.

Types of Data Clusters from the example shown in Figure 8.

- Base64 encoded-JPEG Compressed-Digital Video,
- JPEG Compressed File Size (IMAGE_SIZE),
- GPS,
- IMU, and
- Compass.

Cluster number 1 is from a process. The digital camera generates 15 video frames per second and each video frame has 640 x 480 16-bit pixels. The data rate from the digital camera was too high to capture directly so it was compressed with a JPEG compression process. To put the data in an XML data stream we had to convert the binary data to text with a Base 64 encoding process. The data in Cluster 1 represents digital video data that has been JPEG compressed then Base64 encoded. At this point is just one big text blob. When the processor receives this data it will know to un-Base64 encode then un-JPEG compress, in that order to get to the video data. The JPEG compression process generates varying size of files depending on the entropy of the image data. This process will generate a value indicating the size of the frame. We chose to handle this value as a separate sensor reading. We could have just as well incorporated it into the Base64-JPEG-Video cluster as two data Units, a length value, and a binary blob.

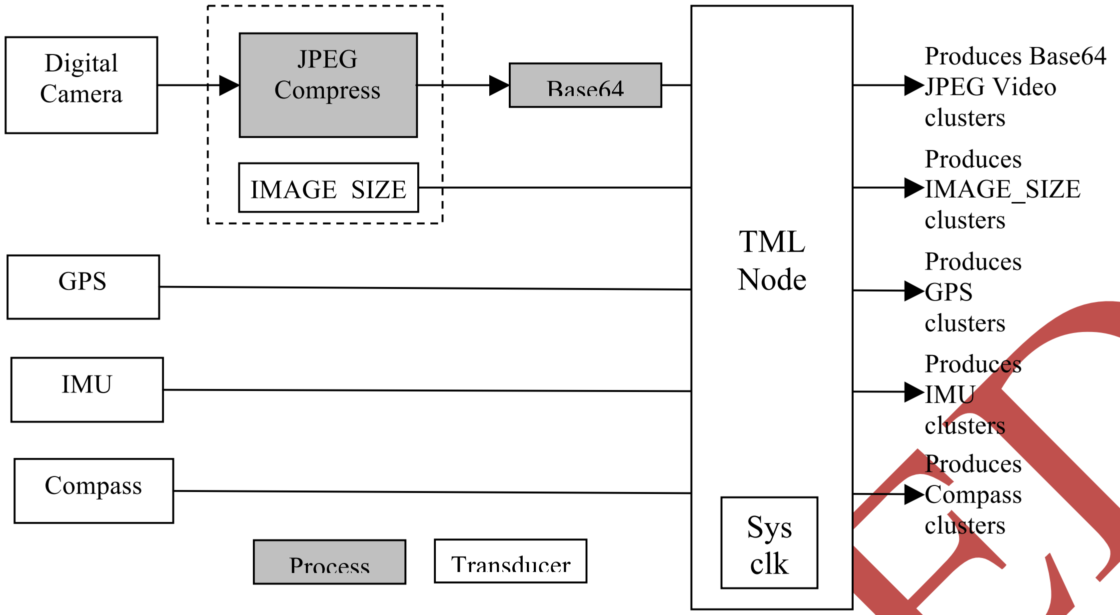


Figure 8 - Example TML system

The TML data stream contains the changing values from the individual transducers. The following example XML data is an example of streaming data from the example system shown above. Each <data> element contains a *data* cluster from the respective transducer or process. The ref attribute in the <data> start tag contains a reference to the description of the data contained in the element. In the following example, the five sensors described above are streamed in chronological order. The following stream described the state of the *transducer system* over time. In some situations, data may come out of time order. TML processors should be aware of this and buffer data long enough to account for this. Figure 9 shows how TML data could be exchanged over a network.

```
<data clk='28118774' ref='IMU'>22.8,1.1,3.4</data>
                              <!—IMU: true heading, pitch roll-->
<data clk='28118792' ref='COMPASS'>21.1</data>
                              <!—COMPASS: mag heading-->
<data clk='28118795' ref='GPS'>516866,-4702126,4264297.2005-08-26T16:31:49Z</data>
                              <!--GPS: X, Y, Z, Time-->
<data clk='28118800' ref='IMAGE_SIZE'>49094</data>
<data clk='28118800' ref='CAM'>...base64 JPEG video...</data>
<data clk='28118874' ref='IMU'>23.9,2.7,-1.1</data>
<data clk='28118888' ref='Weather'>0,15.3,35,18.8,18,0.0,22.5, 82.3</data>
   <!--WX: rain,dewPt,humid,temp,wndchl,wndSpd,wndDir,baroPres-->
<data clk='28118899' ref='IMAGE_SIZE'>49388</data>
<data clk='28118899' ref='CAM'>... base64 JPEG video...</data>
<data clk='28118974' ref='IMU'>0.1 -1.2 1.1</data>
<data clk='28118999' ref='IMAGE_SIZE'>49252</data>
<data clk='28118999' ref='CAM'>... base64 JPEG video...</data>
<data clk='28119174' ref='IMU'>-0.1 1.3 -0.2</data>
<data clk='28119199' ref='IMAGE_SIZE'>49628</data>
<data clk='28119199' ref='CAM'>... base64 JPEG video...</data>
<data clk='28119227' ref='COMPASS'>22.5 0.2 -1.2</data>
```

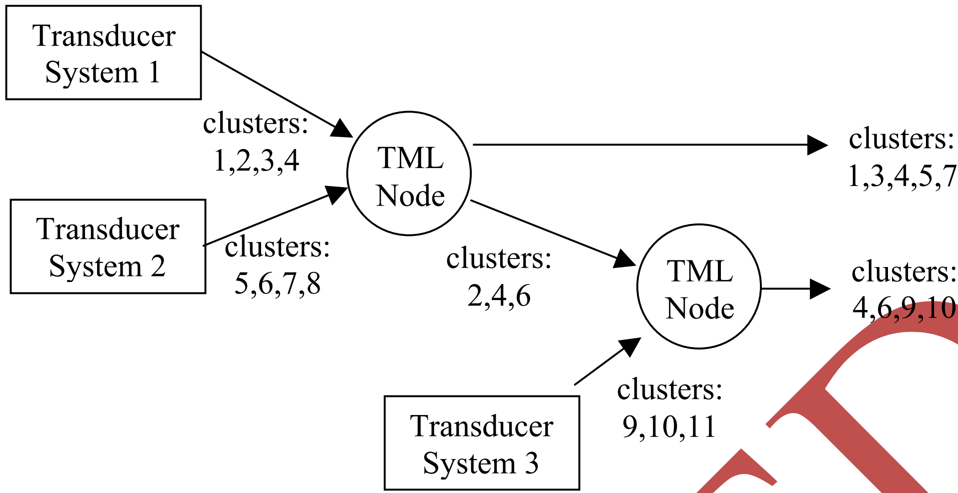


Figure 9 - Merging and splitting of data clusters through network nodes

As data from transducer systems is published on the internet, it can be split and merged to meet individual user needs. TML network nodes will manage the splitting and merging of clusters and maintain the data description documents such that they adequately reflect the data. As new systems come online, there will be a need to update the data description. The TML stream may, from time to time contain updates to the customized data description of split and/or merged streams.

Issue Name: tml rwg 1.19 (S. Cox, 3 Oct 2006)

Issue Description: A "data stream" is effectively a special case of ISO 19123 CV_DiscreteCoverage, where the "geometry" property of the element/CV_GeometryValuePair is constrained to have type TM_Instant (see ISO 19108). The various encodings discussed in the rest of the document are _implementations_ of a "DiscreteTimeInstantCoverage" datamodel. Linking the discussions to this abstract data-model is an important connection to the standards baseline

Resolution: Identify as an issue and document as future harmonization work with ISO Specifications (S. Havens, 3 Oct 2006)

7.14 Asynchronous sampling

A system may employ multiple transducers to observe or take action on multiple phenomena at the same time. Because not all phenomena have the same natural frequency, the different transducers will sample at different rates determined by the natural frequency of the phenomenon or other factors. To understand what the state of the system of a transducer is at any point in time it is important to time tag the data so that data can be interpolated between updates.

7.15 Sequence and timing of clusters

TML data streams interleave clusters from multiple transducers onto a serial channel. In many multi-transducer systems, the individual transducers are not all triggered to capture their data at the same instance in time. The transducers will trigger at different rates. Figure 10 illustrates a system of four sensors sampling at different rates. To establish what the state of the four transducers (sensors) are at any instant in time their values will have to be interpolated between samples. This can only be accomplished if the times of each sample have been carefully time tagged with a high-resolution clock. It is important that a single clock be used to maintain relative timing relations between all the sensors samples.

The sequence of the clusters may be skewed slightly because of data buffering issues. Time tags in the cluster start tags facilitate in maintaining relative timing relationships of the data clusters in the data stream. Using the time tag will enable the clusters to be properly sequenced at the destination by ordering data clusters by time.

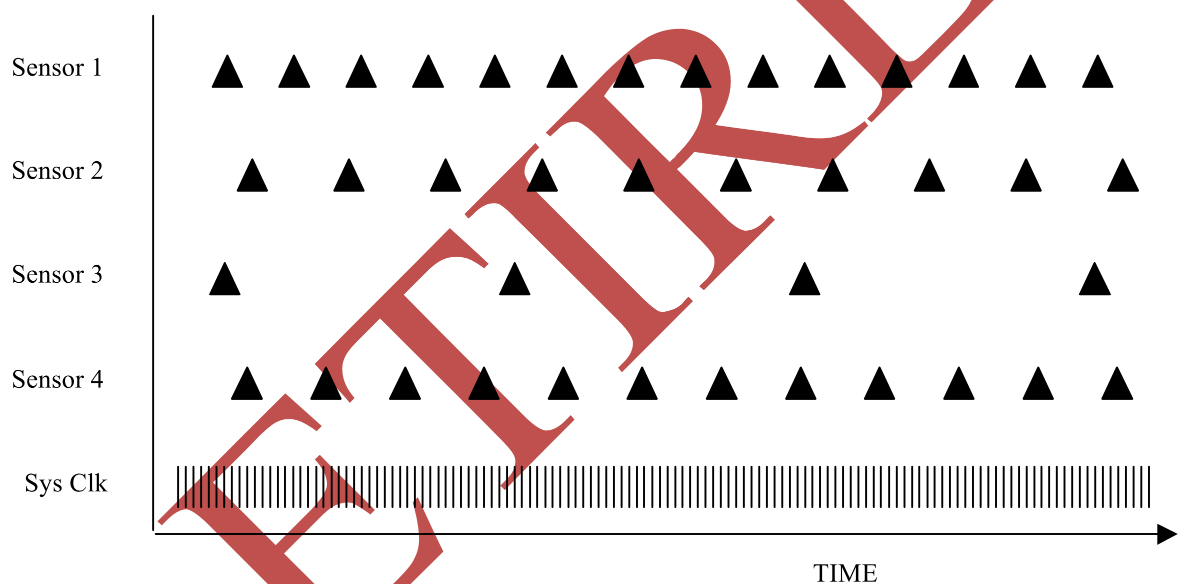


Figure 10 - Unsynchronized transducer triggering

7.16 Verbose and non-verbose time and reference attributes in the data element

The start tag of each data element contains a “clk” and “ref” attribute or a “dateTime” and “reference” attribute. These are relative versus absolute time and reference attributes of the data cluster with the data tag. The “clk” attribute contains the state of the system clock at the first instance of data in the data cluster. The “ref” attribute contains the relative address, which points to the data description for the description of data contained in the data cluster. If there is a need to send the data in a self-contained package, then the full verbose versions of time and reference ID (dateTime and reference) attributes may be used in the start tag. The shortened relative version or terse version is to minimize data overhead in the exchange and archive process.

7.17 Transducer data structure

The transducer data structure is comprised of three levels. The most basic data component is the dataUnit. One or more dataUnits shall compose a dataSet and one or more dataSets shall compose a TCF dataArray. To put Figure 12 into perspective, this structure can be related to an image. The TCF dataArray is the image frame, the dataSet is the pixel, and the dataUnits are the red, green, and blue values. Other types of transducers may not have as complicated of a structure. For example, a spectral radiometer will have a TCF dataArray of one dataSet, and a dataSet of several dataUnits. An even simpler structure is an audio microphone. The TCF dataArray contains one dataSet, and a dataSet contains one dataUnit.

TML data clusters can be transformed into any XML encoding that accurately expresses the above groupings. It may be viewed as a transport mechanism independent of client encodings and primarily concerned with the fusing of data through temporal and spatial correlation. TML data clusters can include data in any meaningful pieces which are reasonable.

A data array has the following structure:

- A data unit is an individual value at the smallest level
- A data set can contain one or more units
- A TCF dataArray contains one or more sets

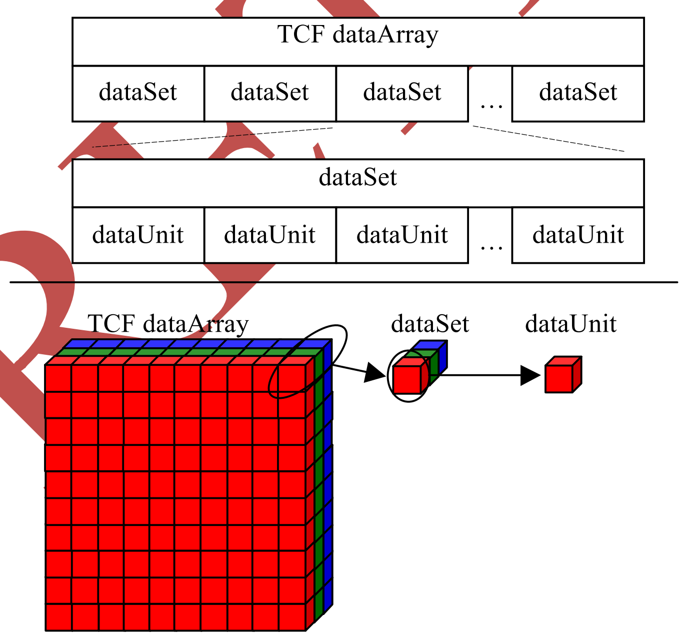


Figure 11 - TML Data Structure

Figure 11 illustrates the TML data structure. The TML data model corresponds to one TCF dataArray, and the definition of a TCF dataArray is based on what is appropriate for a particular transducer. For example, a temperature sensor making single measurement would have as its dataUnit a single measurement with the dataSet composed of one data Unit and the TCF dataArray composed of only a single dataset. In the case of a camera, an individual color component within a pixel would constitute a dataUnit while a pixel within the image would be a dataSet, and the entire image would be its TCF dataArray.

The TCF dataArray when exchanged is a serial chain of data. However, it may have a well-organized data structure of columns, rows, and/or planes. If the data structure is well-organized, then the transducer data structure will be organized giving priority to the spatial distribution of the TCF dataArray cells. The temporal distribution will affect the direction through the sequence and may affect transducer data structure when there is no differentiation in the spatial structure (e.g. radar). In Figure 12 the transducer data structure for the line scan sensor is a single line of ten datasets comprising one TCF dataArray. The TCF dataArray begins with the rightmost scan position at time zero and sequences through to the leftmost scan position at nine time units after. Each cell in the array has a data index number, corresponding to its transducer data structure order. When this array is transported or exchanged, it may be transported out of sequence. If the sequence of the data index number is sequenced exactly as the data is re-sequenced then the transported data structure can be re-sequenced back into the transducer data structure.

A TCF dataAray of 10

data index number	1	2	3	4	5	6	7	8	9	10
ambiguity space κ angle	-1	-.7	-.5	-.3	-.1	.1	.3	.5	.7	1
ambiguity space φ angle	0	0	0	0	0	0	0	0	0	0
offset time	0	1	2	3	4	5	6	7	8	9

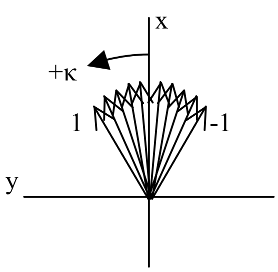


Figure 12 - Data structure for a simple linescan sensor

It should be noted that the sequencing of the structure in the transport might be different than the sequence of the structure as it is defined in the transducer model.

DataSets contain one or more dataUnits. DataUnits within a dataSet typically share similar spatial and temporal coordinates or characteristics. There may be slight internal differences among dataUnits within a dataSet which may be described through offsets. However dataUnits sampled at different rates or dataUnits with very different spatial coordinates should not be mixed within a single dataset.

7.18 Transducer Characteristic Frame (TCF)

A Transducer Characteristic Frame (TCF) is a logical structure applied to transducer data. Although dataUnits, dataSets and TCF dataArrays describe how data is physically arranged and transported, a TCF is a logical structure which describes how data should be interpreted.

As an example, an image may consist of pixels, each of which has a Red, Green and Blue value as described in the previous section. The dataUnits are the Red, Green and Blue values, the dataSets are the pixels, and the TCF dataArray is the entire image frame. Logically, the camera produces frames of data, so the TCF would be the entire image in this case.

Fundamentally, a TCF dataArray can be thought of as an array of n-cells. The number (n) of cells is determined by the number of unique spatial and temporal coordinate values required to characterize one elemental period of the transducer. TCF dataArrays are classified by dimensionality. The TCF dataArray is a key concept for modeling the internal relationships of a transducer. The TCF dataArray enables response and geometry model data to be associated with the transducer data. Transducer data will be captured such that the boundaries of the TCF dataArray are resolvable. There will be a one-to-one correlation between the data collected in TCF dataArrays and the modeling dataArrays. The modeling and the data are closely related and tight relationship should be maintained. The number of spatial coordinates assigned to each dataUnit determines the dimensionality of a TCF dataArray.

If a TCF dataArray from a transducer is sent and the TCF dataArray of the data is not it must be assumed that the data contained in the TCF dataArray is the same as the last TCF dataArray received. For example, a radar transmit chirp waveform could be captured with its time tag on a transmit pulse of a radar. If every transmit waveform from that point on in the stream is the same, then only the clock state of the trigger for that transmit waveform will be in the data stream. It will be assumed that the data from the waveform cluster describing the transmit waveform was in that spot of the clock state.

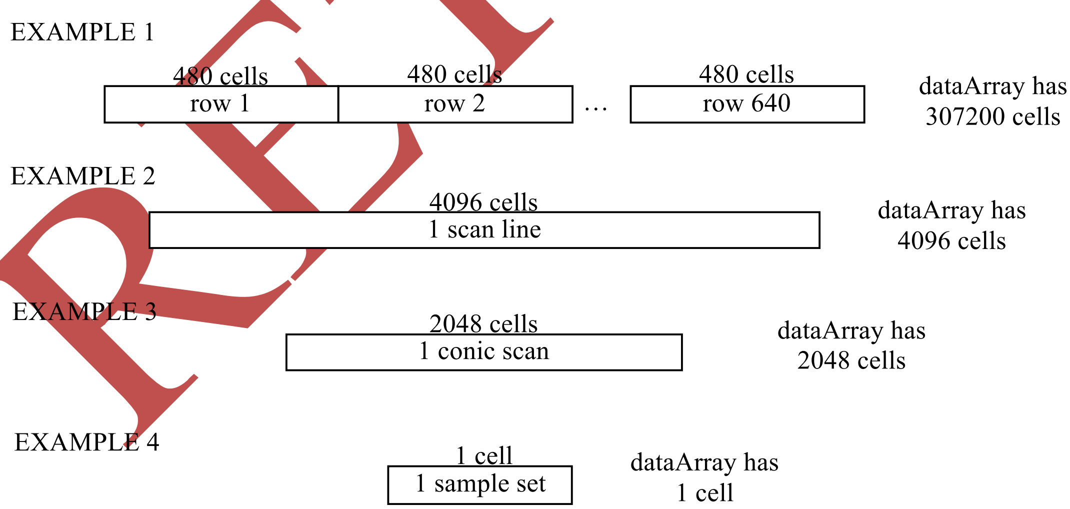


Figure 13 - TCF dataArray examples

EXAMPLE 1: A TCF dataArray for a camera representing one frame with an array of 640 x 480 has 307200 cells in the TCF dataArray, 640 columns and 480 rows. The dimensionality of the array is two (rows and columns).

EXAMPLE 2: A TCF dataArray for a single detector line scan sensor representing one line, with an array of 4096 x1 has 4096 cells in the TCF dataArray, 1 row and 4096 columns. The dimensionality of the array is one (row). .

EXAMPLE 3: A TCF dataArray for a conical scan LIDAR sensor representing one conical scan, with an array of 2048 x 1 has 2048 cells in the TCF dataArray. The dimensionality of the array is one (row). .

EXAMPLE 4: A TCF dataArray for a positioning system that measures its own x, y, & z position relative to a reference, produces a measurement every second, with the x coordinate, y coordinate, and z coordinate of its position. This sensor has a TCF dataArray with only one cell. There is no spatial significance to its position with the array, and there are no spatial coordinates assigned to describe interior geometric orientation, therefore this sensor has a dimensionality of zero (0).

NOTE: All in-situ transducers will have a TCF dataArray dimensionality of zero (0) because they have no ambiguity space.

7.19 Transducer order re-sequencing for transport

TML serializes transducer data in preparation for transport. For example, a four-detector line scan sensor, which produces four lines by 1K pixels long on each swath or spin of the mirror, has a TCF dataArray of 4K pixels or 4K dataSets. The order the pixels are generated as follows: First pixels in each of the four lines are created at the same instant then the second pixels are created in the next instant. Corresponding pixels in the four lines are all sampled in parallel. The transducer logical data structure would be to sequence the pixels in the first line then the pixels in the second line, then the third, then the forth. However, the transporting the pixels may or may not conform to the spatial organization of the pixels. The transport may favor the temporal order of the pixels where it would send the first pixels of each row before sending the second. As you can see, there are multiple ways of transporting data. The resequencing order is defined relative to the logical data structure by defining the index order of each data structure (arrays, sets and units). The data structures can be hierarchical such that interleaving of data across the logical structure components can be achieved. Refer to the examples on the TML webpage.

Issue Name: tml rwg 1.21 (S. Cox, 3 Oct 2006)

Issue Description: Sequencing rules should be compared with CV_SequenceRule described in ISO 19123 clause 8.15 and Annex D. Sequencing in ISO 19123 discusses the sequencing through an array such that attributes in a string can be associated with an array position

Resolution: Identify as an issue and document as future harmonization work with ISO Specifications (S. Havens, 3 Oct 2006)

7.20 TML and binary data

Because XML cannot efficiently carry binary data, as discussed previously, TML enables the binary data to be sent or exchanged via binary means.

7.21 System description

A system description is composed of a description of a transducer and processes as well as data cluster descriptions. The system description also includes information about how the functional components are related to each other.

The data description contains detailed descriptions of the components that comprise a transducer system. The fundamental components are: a system clock, transducers, processes, and relationships. Each of these is described in more detail below.

The transducer descriptions provide basic descriptions of not only transducer data but also how that data is produced and transducer behavior. Process descriptions contain descriptions of the processes and their inputs and outputs. The data cluster description provides the parameters to enable interoperable exchange of transducer data. The various parts of data description documentation (transducer behavior, process behavior, data description and relations between various components of the system) provide the two parts necessary for interoperability and data understanding.

7.22 Common system clock

TML data philosophy requires that when acquiring data from multiple sources the data should be time tagged with a common clock. Using multiple clocks makes it more difficult to correlate data to high precision if required. Figure 10 in section 7.14 illustrates how a common system clock can time tag transducer sampling events so that relative and absolute time relations can be maintained throughout the exchange and archive process. It is possible to provide a precise time stamp for every data unit within a data stream with very little overhead. With proper time stamping and temporal calibration of the transducer the instantaneous state of all transducers can be calculated for any instant in time.

7.23 Transducers

A transducer reference model is shown in Figure 14. This is a very simple model. The transducer is the interface between the data world of computers and the real world. All information and actions between the two domains go through a transducer. A transducer therefore transforms a real world phenomenon into data or vice-versa. TML will characterize the what, where, and when aspects of transducer data. The following sections will discuss this in more detail.

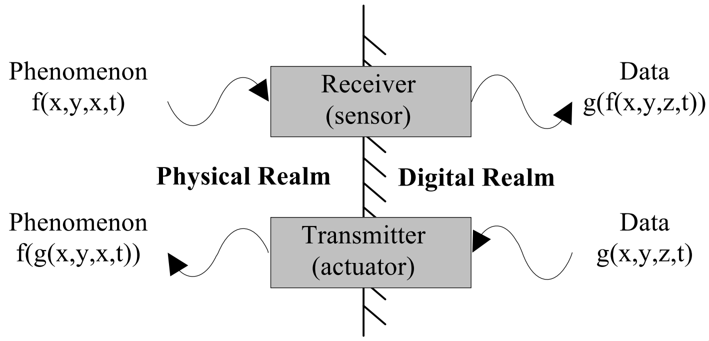


Figure 14 - transducer reference model

7.23.1. Time, space, and value aspect of transducer data

7.23.1.1 Temporal modulation

For computers to handle data from transducers the data must be digitized. Since all transducers are analog devices, their analog signals are converted to a digital signal through an analog to digital sampling process. The frequency of this sampling process is dependent on the highest rate of change of the analog signal. The analog signal is a direct function of the phenomenon signal. To accurately capture the state of phenomenon, the transducer must be able to respond as fast as the phenomenon changes (or is modulated), and the analog to digital sampling process must sample the analog signal at a rate to adequately represent the analog signal in digital form. If analog data is sampled at least at twice its highest frequency the digital data will adequately represent the analog signal in most cases. The rate of data is therefore dependent on the rate at which phenomenon can be modulated. TML characterizes these frequencies and timing considerations of transducers, so that it is known when the limits of the transducer have been exceeded and the data should not be trusted.

7.23.1.2 Spatial modulation

Phenomenon can also be spatially modulated. In other words, the state of a property can vary as a function of location, such as the optical reflectance of the earth's surface. From outer space there is a different reflectance value for every point on the earth. The reflectance value of the optical phenomenon is spatially modulated. Imaging sensors are a class of transducers that detect spatial modulations as well as temporal modulations of phenomenon. Just as the temporal modulation has frequency consideration for transducers so do spatial modulations of phenomenon. There are types of phenomenon where the spatial modulation frequency is too high to be detected by even the best imaging sensors. Transducer spatial characteristics such as modulation transfer functions are required to be known if the resolution of an image is important. There are many forms of imaging transducers. Spatial and temporal modulation transformations of the transducer must be understood before we can accurately model the real world.

7.23.1.3 Value

The accuracy of a transducer response is dependent on the transducer's ability to represent the space and time modulated phenomenon. Besides being able to track the frequencies of change, the transducer must accurately represent the magnitude of the phenomenon. When the temperature goes up one degree, the transducer must not report something different. The transducer must also accurately represent the relative or absolute value of the phenomenon.

To understand the data from a transducer, or to understand what data to give a transducer we must understand the space, time and value aspects of the transducer (or the what, when and where questions).

7.23.2. Transducer modeling concept

Transducer modeling data is necessary to provide a TML application with the necessary information to process TML data from a TML transducer or process system. The transducer characteristic frame is utilized extensively in the modeling of transducers and in defining how transducer data should be interpreted.

7.23.3. Behavior Models

In general behaviour models are useful to any application reading data from another producer or producing transducer data for others to use. Behaviour models include geometry models (which include spatial and temporal models) and response models (which include steady state, impulse, and frequency models).

Normalization information for a transducer is captured and documented in TML. The data itself is not required to be normalized. To illustrate the value of normalization, assume several separate and independent groups are all collecting weather information.

The first group is a weather forecasting team which logs their information into a database. Their desktop client interfaces with a web service which provided temperatures in degrees Fahrenheit. They log these Fahrenheit values and their client displays Fahrenheit values.

The second is a school study group obtaining and logging temperatures in degrees Celsius. Similarly to the first group, this group's client displays Celsius values.

A third group logs thermocouple readings for their temperature data.

If we want to create a single client to interface with these three data sources it would have to know all associated units of measure as well as how to convert them to a common unit of measure. TML simplifies the problem by provides normalization information along with the raw data that is captured.

All behaviour models are applied to the dataUnit, that is, the individual data value representing a particular transducer phenomenon value. Each parameter of the behaviour models is associated with this transducer dataUnit in one of two ways:

- *static* response parameter: an unchanging single value or unchanging single array of values associated with each dataUnit in the data stream. This can be sent once at the start of a data transmission and applied to the data at the receiving end.
- *dynamic* response parameter: a changing single value or changing single array of values associated with each dataUnit in the data stream. Dynamic association requires a separate stream to carry the changing response parameter. This stream is described in TCF format as coming from a virtual sensor

7.23.3.1 TML geometry model

Transducer data geometric characteristics are factors of both spatial and temporal characteristics. Both spatial and temporal characteristics have relative and absolute considerations. The following paragraphs will discuss these relationships.

7.23.3.1.1 Spatial characteristics

To determine where transducer data is (its position) spatial models of transducers are used. Transducer data location is described by both relative and absolute positioning. The absolute position of transducer data is derived from the relative position with the aid of other transducers.

The spatial geometry model and the temporal model enable the mapping of data to real world spatial coordinates or ambiguities whichever the case may be. The real world geometry relations of data are described using interior or relative geometry, and exterior or absolute geometry. A geometry model is a collection of values that describe the relative space-time relationships of each data unit, in a transducer TCF array, relative to a coordinate reference systems.

A primary capability of TML is to facilitate the precise and accurate geo-positioning of transducer data. This enables the rapid space and time association of transducer data from disparate transducers as well as with other geospatial data (e.g. maps, DTED). To this end, TML goes to great lengths to maintain relative and absolute spatial and temporal measurements for each transducer data unit, as well as describing the logical relationships of multiple transducers in a multi transducer system.

7.23.3.1.1.1 Internal spatial characteristics

Internal geometry can be described such that it is independent of the location, attitude, date, and time. Once the interior geometry is defined then it can be positioned relative to absolute spatial and temporal datum's. Only remote transducer need to define their internal spatial characteristics. The internal spatial characteristics of transducer data are described by characterizing the ambiguity shape and ambiguity shape position for each data unit.

7.23.3.1.1.1.1 Ambiguity space

A data unit corresponds to a phenomena, the phenomena may be localized (i.e. come from a specific point) or the phenomena may be distributed (accumulation from a field). Only remote transducers have an ambiguity space, it is the volume of space of which the phenomena is known to exist. An ambiguity space will have a shape and a position defined relative to a specific spatial reference system. In some situations, the ambiguity space may be a single point. In other cases, the shape will define some volume of space (line, plane, sphere, arc, cone, cardoid, teardrop, etc.). The figures below show some examples of ambiguity space. The ambiguity shape can represent a surface, a closed or solid shape, or an open or inverse solid. The shapes may be well formed or they may be irregularly shaped such as a terrain model. Ambiguity shapes may be described using any of the coordinates available in the three coordinates systems available in TML (rectangular, spherical, and cylindrical) relative to any of the reference systems (ECEF, Local, Transducer). Typically, the ambiguity shape will be basically the same for every data unit in the TCF. There may be small variances over the array, which can be described by an array variable. The ambiguity space is a space where it is highly probable that the object or feature creating the phenomenon is located. For example, an Instantaneous Field of View of a camera pixel is an ambiguity space for that camera. The antenna pattern for a radar is an ambiguity space for that radar. Transducer data may have multiple ambiguity spaces associated with it. The resultant ambiguity space can be composed of multiple ambiguity spaces of either solid, inverse solid, or thin surfaces. For example after radar data is received and it is associated with the transmit pulse, each dataUnit in the received range gate has a unique range associated with it. It is therefore possible to determine that objects within the transmit and receive antenna pattern and at a constant range (sphere) could contribute to the response.

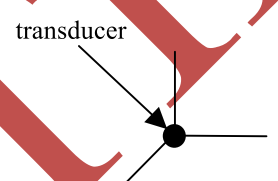


Figure 15 - In-situ transducer (no ambiguity space)

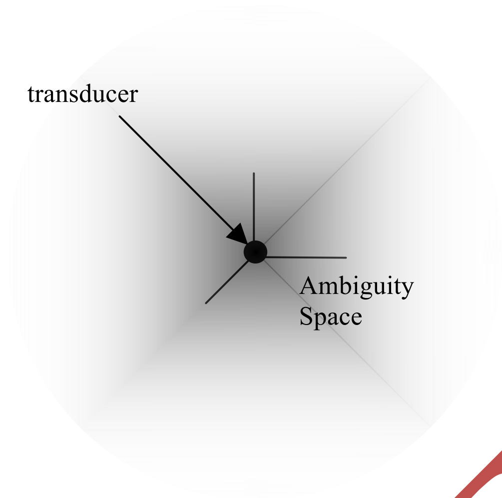


Figure 16 - Remote transducer omni-directional ambiguity space

Figure 17 - Remote transducer directional ambiguity space

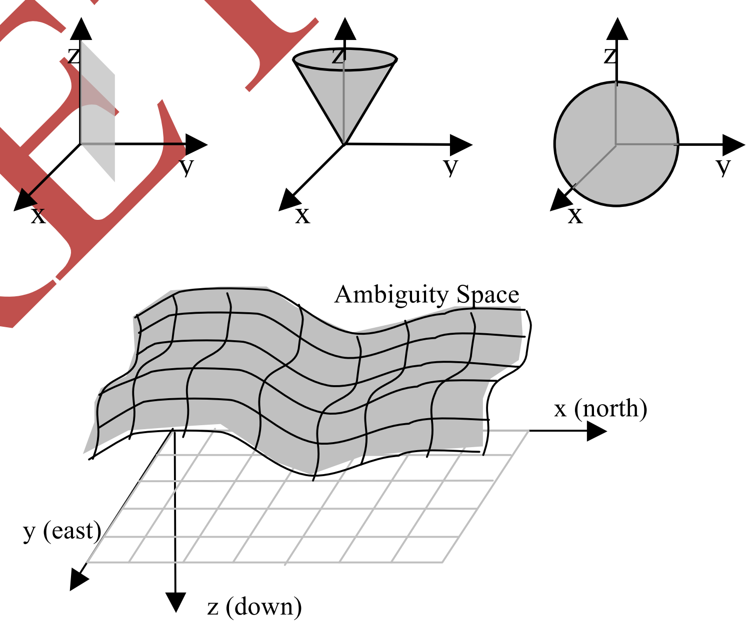
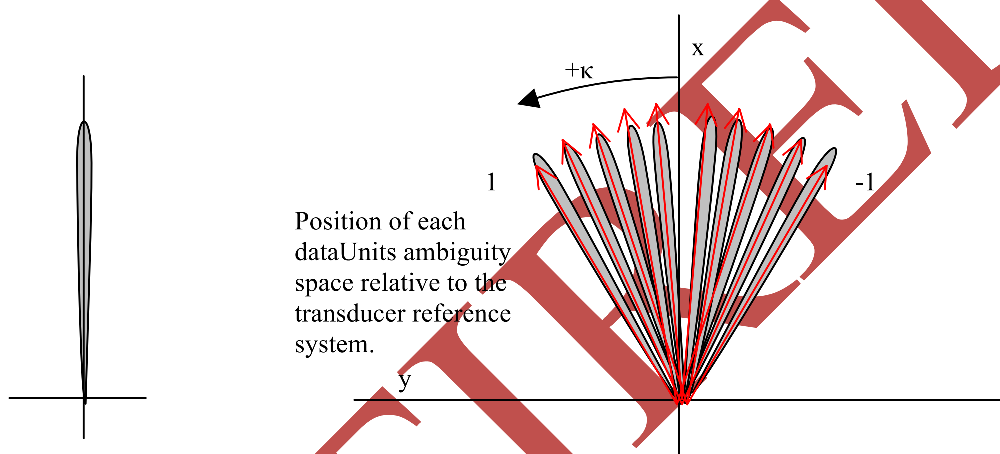


Figure 18 - Examples of geometric and irregularly shaped ambiguity space

7.23.3.1.1.1.2 Ambiguity Space position

The position of the ambiguity shape for each data unit may be different. Figure 19 illustrates the characterization of the internal geometry relative to the transducer reference frame and the start of the scan. In this case, the spatial position of the ambiguity shape is described relative to a transducer reference system and the temporal reference is described relative to the start of the scan. The omega and phi angles define the orientation of each of the ambiguity shapes relative to the transducer reference system. If the reference system is defined properly the positional variances of each of the data units may change only one or two coordinate parameters out of the six possible (three location and three attitude). For example, in this case the scan was in the direction of constant beta. Only one coordinate varied, alpha.



One TCF of 10 dataSets

data index number	1	2	3	4	5	6	7	8	9	10
dataset κ angle	-1	-.7	-.5	-.3	-.1	.1	.3	.5	.7	1
dataset φ angle	0	0	0	0	0	0	0	0	0	0
dataSet time	0	1	2	3	4	5	6	7	8	9

Figure 19 - Distribution of ambiguity shapes relative to a transducer reference frame

7.23.3.1.1.1.3 Transducer Reference System

The transducer reference systems axes are defined with respect to what makes it easiest to describe the spatial distribution of the transducer data. The transducer reference system is chosen to accurately represent the position of the ambiguity space for each data unit within the data array. The transducer reference system should correspond to the transducer reference system described in the physical system description. The physical reference system description will indicate where the origin of the axis is in relation to the transducer device and the orientation of the principle axis. In some cases, the transducer spatial reference system may move relative to the physical reference system. In this situation, sensors are required to monitor its movement relative to the physical reference. Objects in the transducer spatial reference frame can be positioned with either a rectangular or a spherical coordinate system.

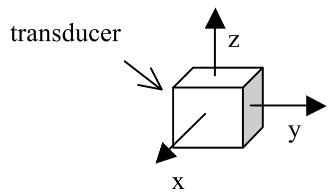


Figure 20 - Transducer spatial reference system

EXAMPLE: The transducer reference system for a camera may be chosen such that the origin of the coordinate axis is located at the frontal node of the optics and the x-axis points along the focal axis of the lens, with the y-axis pointing out the right side of the camera aligned with the horizontal scan of the camera raster. In this manner, each column in the focal plane array will have a common α angle, and each row in the array will have a common β angle. Giving each pixel in the focal plane a unique α and β spherical coordinate relative to this transducer reference system.

NOTE: For most transducers the position and orientation (attitude) of the transducer reference system is important. In some instances, only the position or orientation of the reference frame is important. See location and attitude invariant transducers.

7.23.3.1.1.2 External spatial characteristics

Figure 21 illustrates the exterior geometry of the line scan sensor. The interior geometry is positioned (location and attitude) relative to an external absolute reference system (in this case the earth reference system).

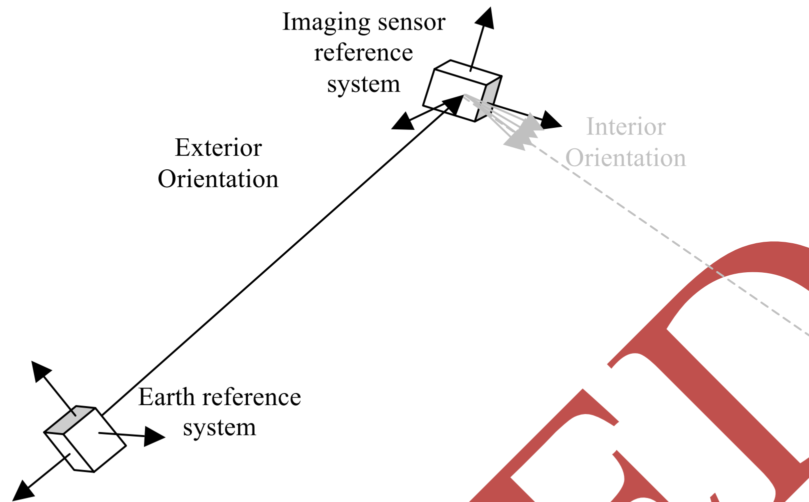


Figure 21 - External geometry relative to absolute reference system

Transducers may be stationary, portable, or mobile. In the case of stationary transducers, their position can be measured and recorded and it will remain the same as long as it does not move. Portable sensors are much the same, they can be surveyed in position, and their position reading will remain unchanged. For mobile transducers the external geometry may not be determined directly in many cases as in the case of the stationary and portable transducers. The location and attitude of mobile transducers may be measured by several transducers working together. As in the case shown in Figure 22, the imaging sensor is positioned using a chain of location and attitude measuring sensors. To determine the external position of the imaging sensor, at any instant in time, the state of each of the sensors responsible for calculating its position (of the imaging sensor) must be determined. This is a reason for applying precise time tags to sensor data.

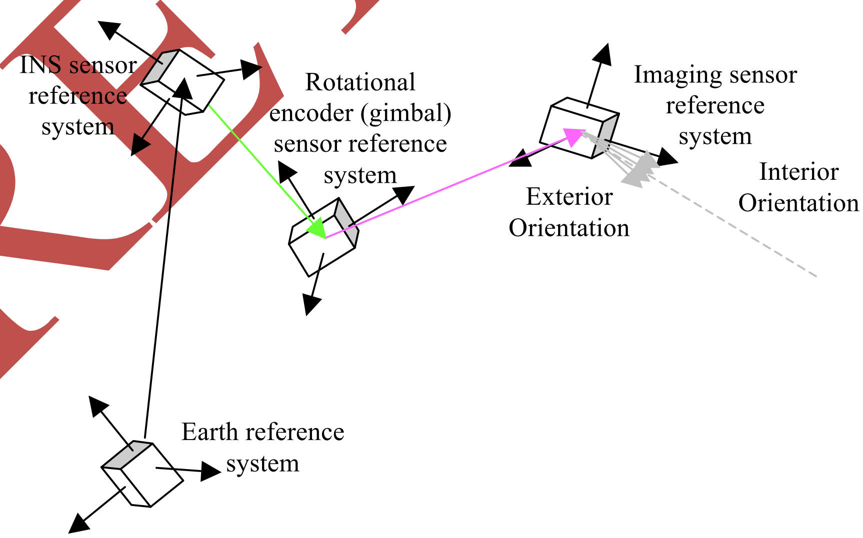


Figure 22 - External geometry through several transformations

7.23.3.1.1.3 In-situ transducers

In-situ transducers sample their data at the site of the transducer. Measurements from an in-situ sensor or actions by an in-situ actuator cannot be applied to any other location around the in-situ transducers.

Only exterior orientation is required to describe the position and orientation of the in-situ transducer. In-situ transducers have no ambiguity space, but in some situations in-situ transducer data may be able to determine information about a location other than the transducer location by knowing details and properties of the feature or medium where the transducer is located. An in-situ transducer may be either attitude or location invariant as well.

EXAMPLES: in-situ receivers - compass, temperature probe, displacement or velocity sensors; in-situ transmitters – stepper motor, heater, mode switch.

7.23.3.1.2 Temporal characteristics

Just as important as determining the spatial location of the data, is determining temporal properties of the data. In many cases, the time of data capture is needed to determine the spatial location of the data. The temporal and spatial characteristics work together to describe an overall geometric characterization of a transducer system. The TCF dataArray describes the internal temporal characteristics for a transducer. The internal temporal characteristics also utilize the relative time reference of the TCF dataArray trigger or the time of the first data in the TCF dataArray. The time of the first data is t=0.

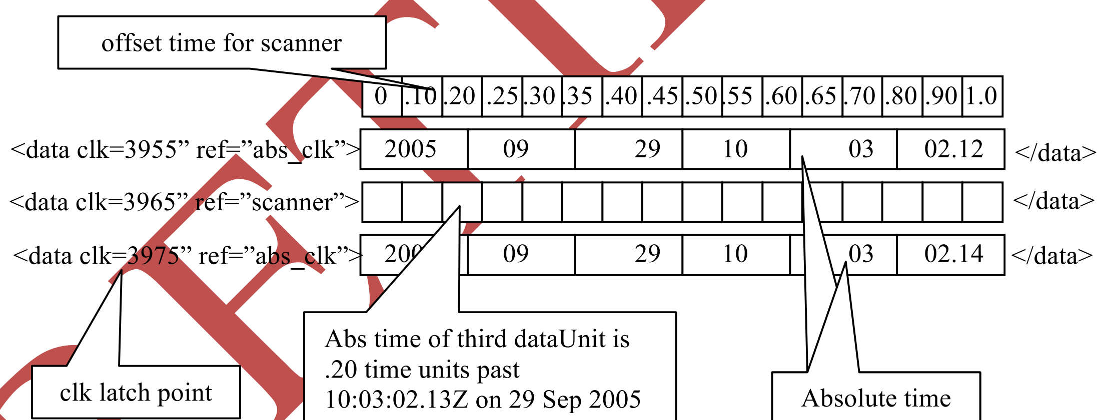


Figure 23 - Internal (relative) and external (absolute) time

7.23.3.1.2.1 Ambiguity Time (Integration time)

The ambiguityTime describes the elapsed time during which the transducer is acquiring information or applying an action. For a sensor it is the time during which the sensor actually samples the phenomena. During this time any modulation of the phenomena is not uniquely distinguishable.

An integration time period (ambiguity time) is the time required by a receiver to accumulate a reading or the time for a transmitter to apply an action. Time resolution can be determined no finer than the ambiguity time. For example, when a camera shutter opens for 1/30 of a second that is the integration time for that set of measurements. When an aircraft flies, an air sampler opens the sampling instrument at 3pm and closes it at 5pm, and then 2 hours is the integration time for that sensor. Any modulation that occurs within the integration time cannot be determined. One value corresponds to response or stimulus depending on if the transducer is a receiver or transmitter respectively.

EXAMPLE: When a camera shutter opens for 1/30 of a second that is the integration time for that set of measurements. When an aircraft flies, an air sampler opens the sampling instrument at 3pm and closes it at 5pm, and then 2 hours is the integration time for that sensor. Any modulation that occurs within the integration time cannot be determined. One value corresponds to response or stimulus depending on if the transducer is a receiver or transmitter respectively.

7.23.3.1.2.2 Latency time

The latency time is the elapsed between the a/d sampling event of the transducer and the time data is available from the transducer. There is only one latency value for a transducer, which correlates to the time it takes for the phenomena to propagate through a transducer, relative to the clk latch point. Figure 24 illustrates the time definitions.

7.23.3.1.2.3 Offset time

In multi dataset Transducer Characteristic Frames, each dataset within the Characteristic frame may have a slightly different time of sampling, For example a line scan sensor scans captures data on one side of the scan before the other. The relative time difference when each sample is captured relative to the first is the offset time. This may also be referred to as internal time. The internal geometry of a sensor is characterized by both internal temporal (offset) and spatial characterizations.

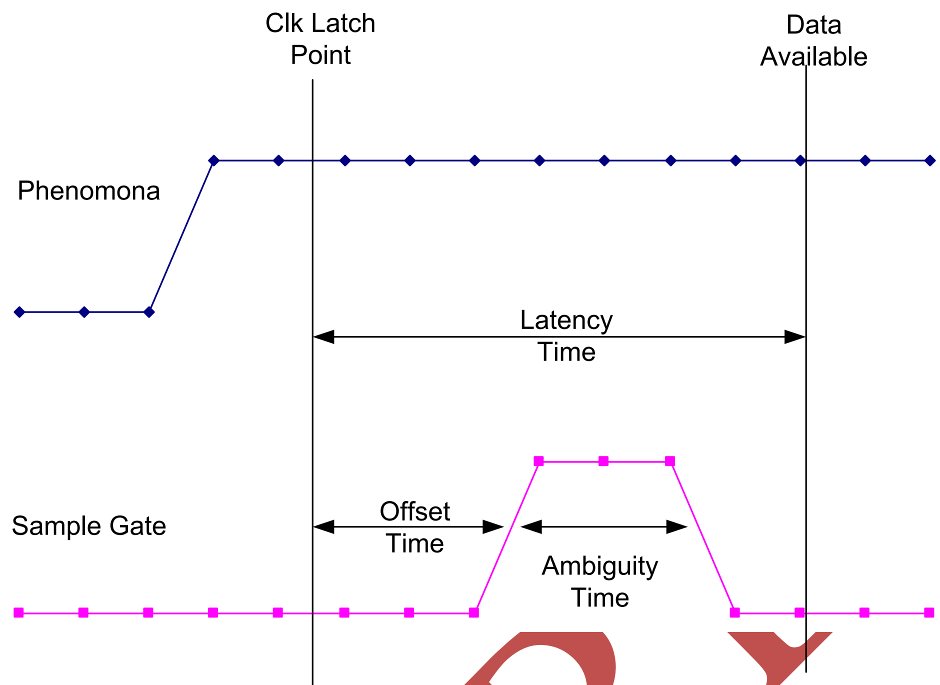


Figure 24 - Time definitions

7.23.3.1.2.4 Transducer events and triggering

An analog to digital conversion of an analog phenomenon into a digital data representation of that phenomenon (or vice-versa) is a transducer event. Transducer events can occur at millions of times a second or transducer events can happen once a century. The rate at which transducers sample or trigger is dependent on many factors. Implied clock state sensors measure the state of the system clock at the TCF dataArray trigger. The clock state sensors value is used in the start tag of the data cluster which contains the data sampled from that trigger event. The following figure is an illustration showing the notional concept of the sysClk, triggers, and clock state.

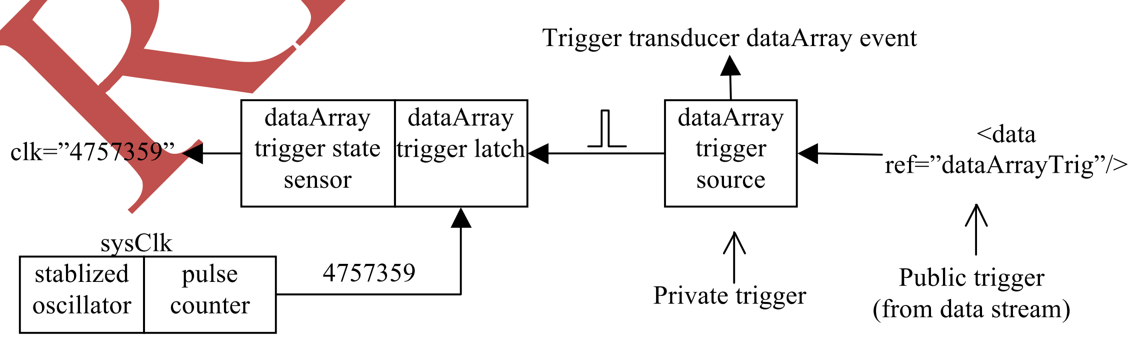


Figure 25 - Transducer private and public triggering and latching clock states

7.23.3.1.2.4.1 Continuous private (internal) triggering

Many transducers sample data continuously, with their analog to digital conversion circuits continually being triggered to take a digital sample of the analog state of the phenomenon. These internal triggering circuits typically trigger the A/D circuits at a periodic rate. This rate is dependent on the frequency content of the analog signal as discussed above. For every sample a transducer makes it outputs a digital piece of data (dataUnit).

7.23.3.1.2.4.2 Random private (internal) triggering

Some transducers vary on random internal triggering. Instead of sending data on every A/D trigger, these transducers only send data when data changes or when data reaches predetermined thresholds. This type of triggering is used with state interpolation. As long as the data does not change, no data is sent. When the data changes it is sent.

7.23.3.1.2.4.3 Public (Remote) triggering

Other transducers will not invoke an action or take a sample until instructed to do so by an external trigger. The TML system and transducer descriptions enable the description of remotely triggered transducers. When a data element is sent in the data stream that corresponds to the triggering actuator, the transducer system is instructed to trigger the appropriate transducer.

7.23.3.2 TML response model

The response model characterizes a transducer's behavior with respect to specific defined inputs. For calibrated receivers (sensors) the response model is used to provide calibration parameters of the transducers behavior such that the phenomenon can be accurately determined from the data or visa-versa. For transmitters the response model is used to determine what the transmitter's output (its action or phenomenon it produces) is going to be based on the data sent to it. Factors that influence the response model are frequency response, impulse response, and steady state response.

7.23.3.2.1 Function modeling

Functions are used throughout TML to describe things such as frequency response and transfer functions. The function coordinates maybe described normalized, however it is not necessary. To convert the normalized coordinates to actual coordinates they must be associated with a multiplier and an offset. For example a transfer function that maps phenomenon to data may have normalized phenomenon function values of [0,.20,.40,.60,.80,1] and normalized data function values of [0,.10,.33,.65,.85,1]. If the phenomenon had a mult of 100 and an offset of 273, and the data had a mult of 255 and an offset of zero then the actual phenomenon function coordinates would be [273,293,313,333,353,373], and the actual data coordinates would be [0,25,84,166,217,255]. Figure 29 illustrates an example of how the normalized coordinates with the associated mult and offset factors are used to derive the actual function. The function may be a dynamic function where it is changing as a function of time. If this is the case the coordinates themselves or the mult and offset parameters may be read and replaced by sensor readings from the data stream.

7.23.3.2.2 Frequency Response

The frequency response is a characterization of how a transducer responds to the various frequency components of the stimulus. TML will use the response amplitude versus frequency for three types of frequency related functions: carrier frequency response, modulated frequency response, and power spectral density. The example in Figure 26 shows how a low pass response of a particular transducer would apply to the carrier or modulated response. When the phenomenon contains frequencies above the cut off point, the response is not able to track it and it will be averaged.

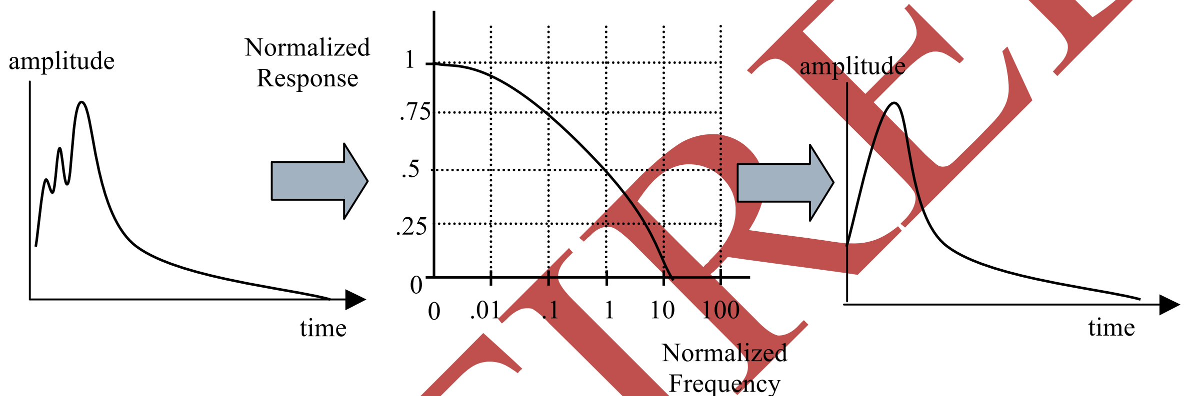


Figure 26 - Low pass response of a transducer

A frequency response defines the transducer's ability to respond to various frequency inputs (modulated frequency) and the sensitive part of the frequency spectrum on which the transducer is responsive (carrier frequency).

Carrier frequency response represents the spectrum of carrier or frequencies which a sensor is sensitive to or is able to detect or respond to. TML expresses the functions as normalized curves, and then applies mult and offset factors to the axis thus modelling the actual function. The frequency of some information is so low that the baseband signal of the information will not propagate through space. To communicate this information it can be carried by a higher frequency signal that will propagate through space. This higher frequency signal is called a carrier signal. The higher frequency carrier signal is modulated by the lower frequency information.

Modulation frequency response represents the spectrum of modulation frequencies which a transducer or process is sensitive to or is able to detect or respond to. TML may express the functions as normalized curves, and then applies mult and offset factors to the axis to model the actual function. The modulation signal is a lower frequency signal that represents information that is put onto a higher frequency carrier signal.

Power Spectral Density refers to the spectrum of frequencies contained in a signal, baseband or modulated carrier. For transmitters this is the power spectrum of the emitted phenomenon. The spectral density of the wave, when multiplied by an appropriate factor, will give the power carried by the wave per unit frequency. This yields the power spectral density (PSD) of the signal. It is usually plotted as power –vs.- frequency. The units of spectral power density are commonly expressed in watts per hertz (W/Hz). TML expresses the functions as normalized curves then applies mult and offset factors to the axis thus modeling the actual function. (See carrier frequency response and modulation frequency response)

7.23.3.2.3 Impulse response

The impulse response is useful for characterizing linear time invariant (LTI) transducers. If the transducer's response in known for a unit impulse input, then the transducer's response can be derived by convolving the input signal with the unit impulse response. Since linear time invariant systems are reversible, given the output signal, then the input signal can be derived through the reverse process. Figure 27 shows an example of a normalized impulse response.

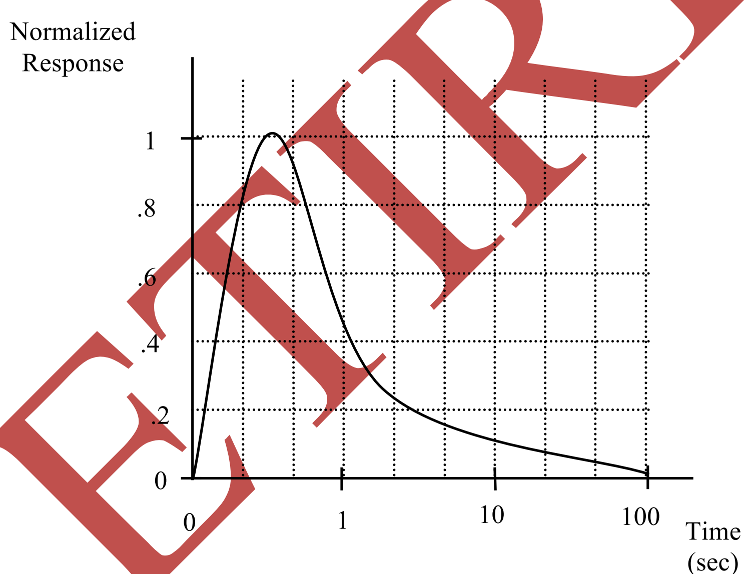


Figure 27 - Example of a normalized impulse response function

7.23.3.2.4 Steady state transfer function

The steady state transfer function provides a mapping of the physical property (phenomenon) to the data for the steady state. This function does not characterize dynamic characteristics of the transducer. This function does however characterize linear and non-linear regions of the transducer. To illustrate the steady state function Figure 28 shows a transducer model that senses the temperature property of the thermal phenomenon.

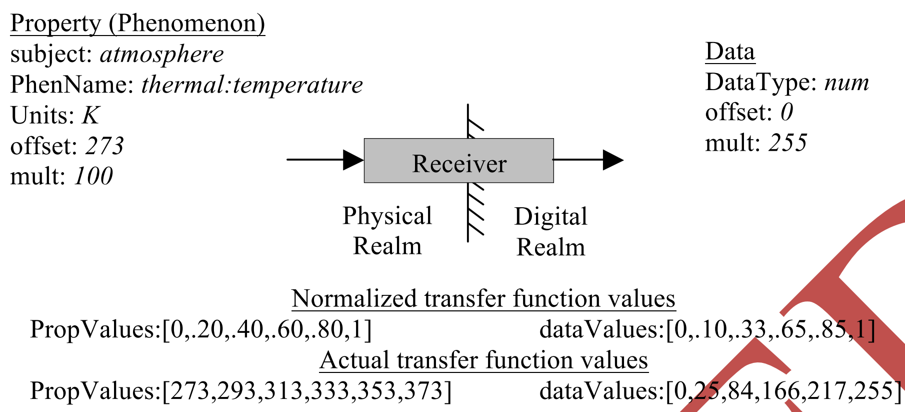


Figure 28 - Model of temperature transducer

A normalized steady state transfer function describes the mapping from property to data. The normalized coordinates given in the model must first be multiplied and offset the proper amount to represent the actual transfer function. In this example, the normalized property values were multiplied by 100 then offset by 273 to derive the actual phenomenon scale. The normalized data was multiplied by 255 with no offset to derive the actual data scale. Figure 29 shows this process.

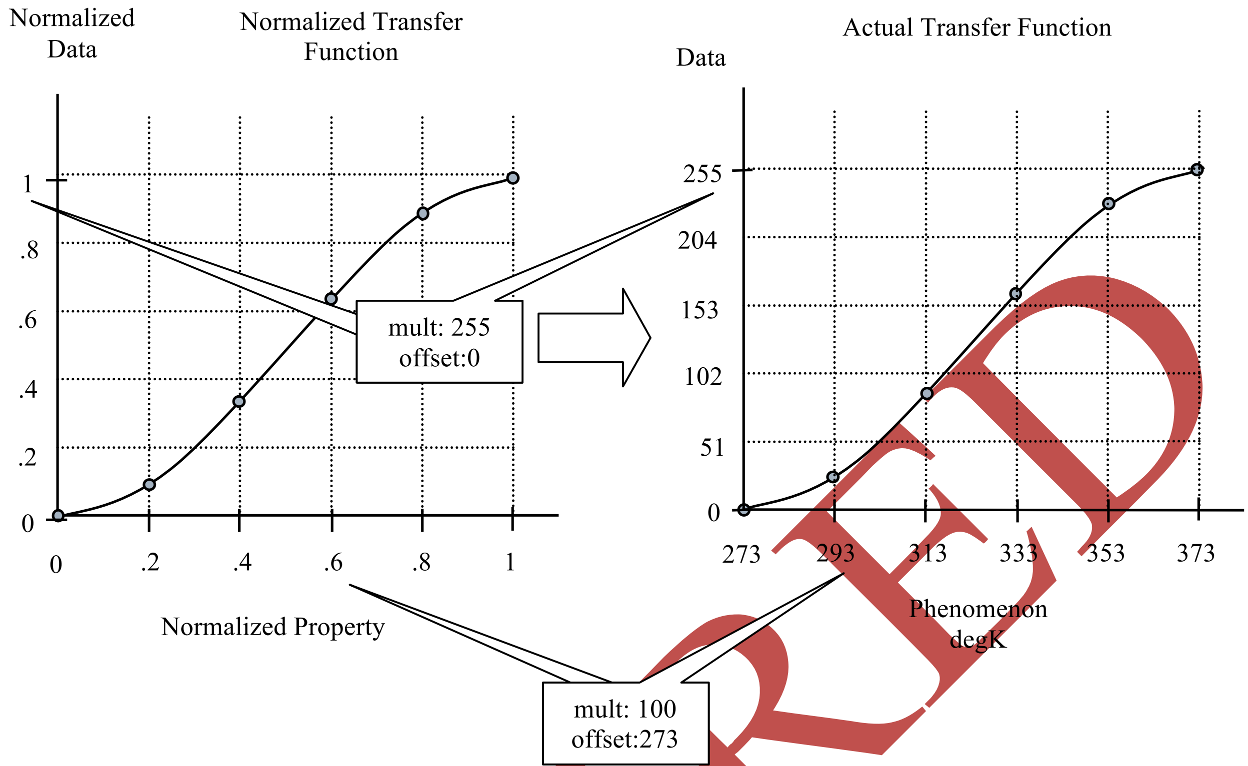


Figure 29 - Deriving the actual transfer function from a normalized function

Some transfer functions use data other than numbers. There are cases where the values represent quantities other than numeric values. The quantities may correlate to a single property value or a range of values. Examples of non-numeric quantities are Empty, Full, Small, Large, Left, Right, High, Low, etc. The following is an example of a transfer function listing using non-numeric data values.

Normalized transfer function values

propValues:[0-.40,.41-.80,.81-1] dataValues:[“low”,”medium”,”high”]

mult: 100, offset: 273

Actual transfer function values

propValues:[273-313,314-353,354-373] dataValues:[“low”,”medium”,”high”]

Event sensors are sensors that listen for a specific event to happen or a threshold of a phenomenon to be exceeded; they will then output a data value. These special event sensors have a non-linear transfer function. For example, a sensor which responds to an over threshold event and whenever the normalized phenomenon value gets above .4 the data output goes high, and when the value drops below .4 the value goes low. Note that the switching point can be different when the phenomenon is increasing than when it is decreasing, and this is known as hysteresis.

Hysteresis is a property of a steady state transfer function where the function is different depending on the direction data is moving in. The following transfer function illustrates a transducer exhibiting hysteresis.

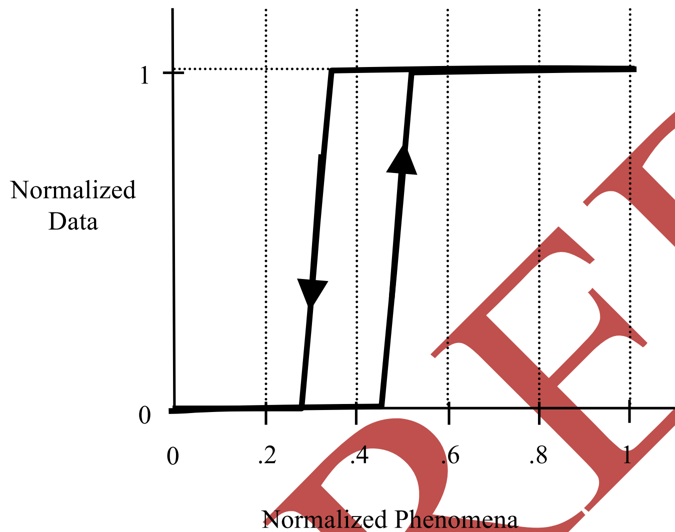


Figure 30 - Transfer function showing hysteresis

The transfer function only shows the range of phenomena that the data range covers. In some instances, the transfer function is dynamic. The transducer may adjust its gain or attenuation of the phenomenon to stay within the dynamic range of the transducer. In these situations, the mult and/or offset factors shall be tracked with a sensing transducer to provide the transfer function scale factors at any instant in time.

A transducer may have more than one steady state transfer function. The transfer function may be different depending on the direction the phenomenon values are going (increasing or decreasing). This hysteresis is characterized separately in each direction.

In addition some transducer may not be calibrated.. In other words there response is only proportional or inversely proportional to the stimulus.

7.23.3.2.4.1 Interpolation of phenomena state in between transducer events

TML captures transducer events that can occur periodically or randomly. These events represent the state of the phenomena at specific points in time. To determine what a transducer state is in-between events it is necessary to know something about the behavior of the phenomena and the transducer itself. TML categorizes this behavior as either continuous, last state, return to zero. Each of the three behaviors is described below.

7.23.3.2.4.1.1 Last state interpolation

The last state interpolation is used when the data to or from a transducer is only sent when a transducer value changes (sensor) or is required to change (actuator). This requires less bandwidth for communication.

```
<ambiguityTime>0</ambiguityTime>

<data clk="00">3</data>
<data clk="11">2</data>
<data clk="39">1</data>
<data clk="44">3</data>
<data clk="52">4</data>
```

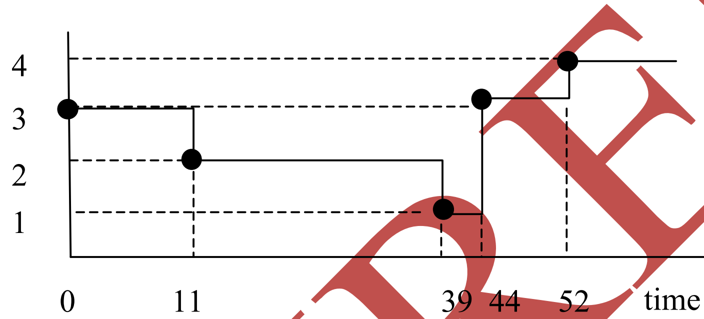


Figure 31 - State (random) events

7.23.3.2.4.1.2 Return to zero interpolation

The return to zero interpolation indicates that the transducer is only measuring or acting on something during the TCF of the transducer event. It can be assumed that the phenomenon is zero in between events. The return to zero may work with either periodic or random transducer events.

```
<ambiguityTime>2</ambiguityTime>

<data t=09>3</data>
<data t=29>4</data>
<data t=39>2</data>
```

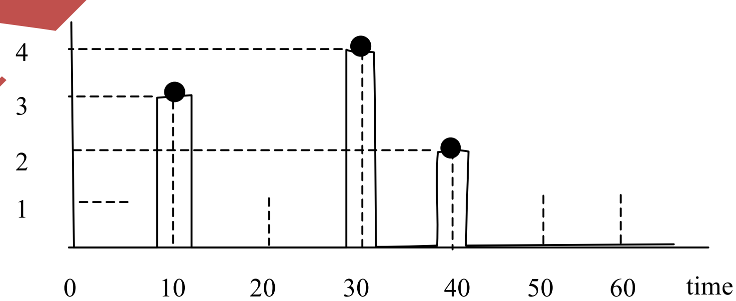


Figure 32 - Return to zero events

7.23.3.2.4.1.3 Continuous interpolation

Data interpolation described as continuous means that it can be assumed that data follows a smooth continuous path in between transducer events. Data in between events may be interpolated with any number of interpolation algorithms, depending on the accuracy desired.

```
<ambiguityTime>0</ambiguityTime>

<data clk=”00”>3</data>
<data clk=”10”>3</data>
<data clk=”20”>2</data>
<data clk=”30”>2</data>
<data clk=”40”>1</data>
<data clk=”50”>3</data>
<data clk=”60”>4</data>
```

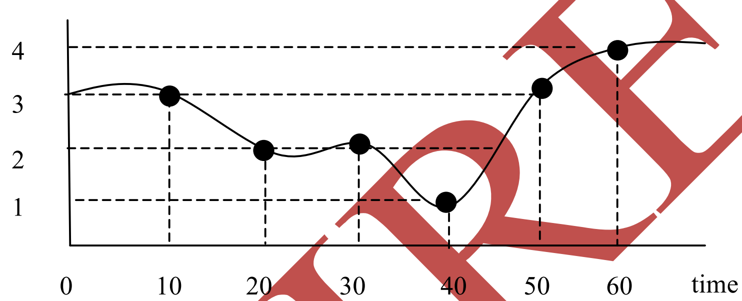


Figure 33 - Continuous events

7.23.3.3 Miscellaneous

7.23.3.3.1 TCF Data Array Values

Parameters of functions or characteristics (such as integration ambiguity time) may be dependent on their position within the TCF dataArray. An example of this is when a CCD array must be radiometrically balanced to obtain uniform intensity response across the array. Position and times of data within a dataSet within a TCF dataArray have coordinates unique to their position within the TCF dataArray. The array values will have the same number of values as cells or indexes in the TCF dataArray. The array values will have a one-to-one correspondence to cells in the TCF dataArray, which lead to the data.

7.23.3.3.2 BindUID

The bindUID is an optional construct for most elements in TML. A bindUID can be a source and/or a sink of data used to bind data at a later time or to link to a dynamic time sequential stream of data. If the data does not need to be transformed (e.g. scale change) for use by the parent of the bindUID then an identity transducer or process can be used. The identity transducer or process does not need to be characterized. The bindUID is treated as a dataUidRef in the cluster description

- Source Example using the element mult: The status of the gain can be read from the Gain Process 1 output

```
<mult bindUid="gain_connection"/>
<dataToDataRelation>
     <source>gainProcess1Output</source>
     <sink>gain_connection</sink>
```

- Sink Example using the element mult: The gain can be changed by changing the input on the Gain Process 2

```
<mult bindUid="gain_connection"/>
<dataToDataRelation>
     <source>gain_connection</source>
     <sink>gainProcess2Input</sink>
```

- Source and Sink Example using the element mult: Sometime a cluster description can describe data connecting directly to or from a bindUID.

```
<mult bindUid="gain_connection"/>
<dataToDataRelation>
     <source>gain_connection</source>
     <source>gainProcess1Output</source>
     <sink>gainProcess2Input</sink>
```

7.23.3.3.3 Relating model data to the transducer data using the TCF dataArrays

TML uses TCF dataArrays to correlate the data to the data descriptions. Transducer data is captured in data clusters and the data cluster is a mechanism for improving the transport efficiency of communicating the array over the communication channel. A cluster will optimally contain one TCF dataArray. However if the array is very large or extremely small then the larger arrays can be split into multiple clusters, or the smaller arrays can be grouped into a larger cluster. Whatever the situation the array boundaries must be readily identifiable.

There are many parameters which are used to describe transducer systems which may be variable by their cell or data index location within the array. Geometry and response characteristics may vary as a function of the cell location within the array. For example, the interior geometric modelling parameters are a type of data description that will utilize the array for containing modelling parameters. The ambiguity shape has a shape and position relative to a spatial reference system. For a case of an imaging camera, this is comparable to saying that a pixel has an instantaneous field of view (IFOV) and each IFOV for each pixel in an image frame has a unique angular offset relative to the imaging sensor reference frame.

Each cell or indexed data location within the array of the geometry model will have a unique coordinate which corresponds to the corresponding cell or data index location in the array used for sending the data in the data cluster. A modelling array is typically constant for a transducer system. In some cases the internal modelling may change. In these instances a virtual sensor will be tasked to monitor the changing parameters. The following figure illustrates the previous discussion. In this figure the modelling data indicates the relative spatial position and relative timing of each of the pixel samples in the ten samples per line linescan sensor. Each scan starts with the first pixel being sampled at the clock time. The first pixel corresponds to -1.0 units kappa angle and a 0.0 units phi angle relative to a transducer reference system. The next pixel is 0.2 time units after the time tag value, a kappa angle of -0.7 units and a phi angle of 0.0 units. The scan sequences through the ten samples or pixels until it reaches the last sample in the line. The last sample is sampled at 1.0 time units after the time tag and is at a spatial coordinate of 1.0 kappa angle units and 0.0 phi angle units relative to the transducer reference system.

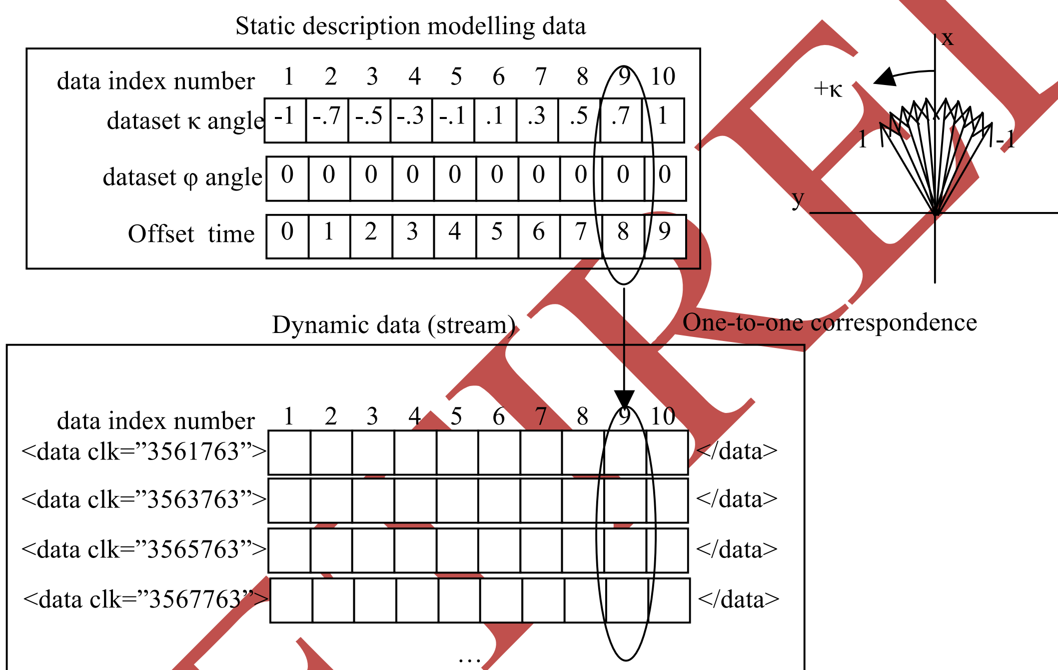


Figure 34 - Using the TCF dataArray to relate modelling data to streaming data

7.23.3.3.4 Precision and accuracy

In the process of measuring and sampling, errors and uncertainty always play a large role. Almost as important as the measurement itself is the uncertainty of the measurement. Every measurement should be qualified with an uncertainty. The uncertainty is characterized by relative and absolute errors from random and systematic sources, and TML provides places to capture accuracy of measurements

7.24 Process components

Transducer data may be processed or may need to be processed for a variety of reasons. When data is received from a transducer the data may have been processed by one or more processes. Alternatively, when data is sent to a transducer it may go through one or more processes prior to the transducer receiving it. It would be ideal if a process could be modeled, such that if the process model and input are known, then the output could be determined. This would be necessary for determining what the data would look like in the transmitter side. It would also be ideal if a process could be modeled such that if the process model and output data are known then the input could be determined. Examples of transducer processes are integration, differentiation, averaging, compression, encoding, frequency mixers, filters, discriminators, etc. A system may be composed of a number of transducers and processes. Each transducer and process is described as a stand-alone entity or component. These components can be reused in other systems as deemed necessary.

7.24.1.1 Process Modeling

TML supports process modeling of vectors, signals, images, and data as well as their properties such as linearity, time invariance, memory, reversibility, etc. Currently, TML modeled processes can only have a single output. Multi-ended processes must be modeled with separate processes. TML models processes by supplying a unique URI identifier, a name, identifying what the inputs and outputs are, and describing any parameters which are unique to that particular process. The URI may point to a site on the network where the process algorithm is defined in more detail.

7.25 Systems relations

The systems relations data takes a higher-level view of the system. The transducers and processes in the system are treated as elements linked by logical and physical relationships. The relations data describes relationships among the various parts in the system as well as to external references such as geographic datums. One of the keys to sensor fusion is to have data represented in a common datum. TML provides a way to capture the linkages of data to standard, space, time and measurement datums.

The transducer and process descriptions are descriptions that an original equipment manufacturer can develop at the time of manufacture. The system relationships description is usually developed after the transducer and process components have been integrated into a system. When the TML document is developed for the system the process and transducer components can be plugged into the system and the relations relationships can be added without changing the previously developed component descriptions.

If we are to understand transducer data and associate it with other transducer data, then the data must be put into context. We must understand basic semantic relations of the data such as what is the data, and where is the data. Data from/to transducers have no context other than relating to a property or a phenomenon. Data to/from transducers can be given context through relations. Relations can be static (non time varying) or dynamic (time varying).

Subjects are things (matter or energy) in the real world. They exhibit their qualities and quantities through properties of phenomena. Sensors are able to detect the phenomena properties and processors and data exploiters are able to make predictions of what the object is by interpreting the phenomenon. The more phenomena from a subject the better the prediction of what it may be. TML handles the complexity of a many-to-one or a one-to-many relationship between transducer data and subjects.

In some cases the subject is known at the time of system development. The phenomenon can then be associated with a subject at that time. For example, a pressure sensor may be installed in a pressure boiler. The phenomenon property of pressure would relate to the subject “pressure boiler”.

The following figures represents the relationships characterized by TML.

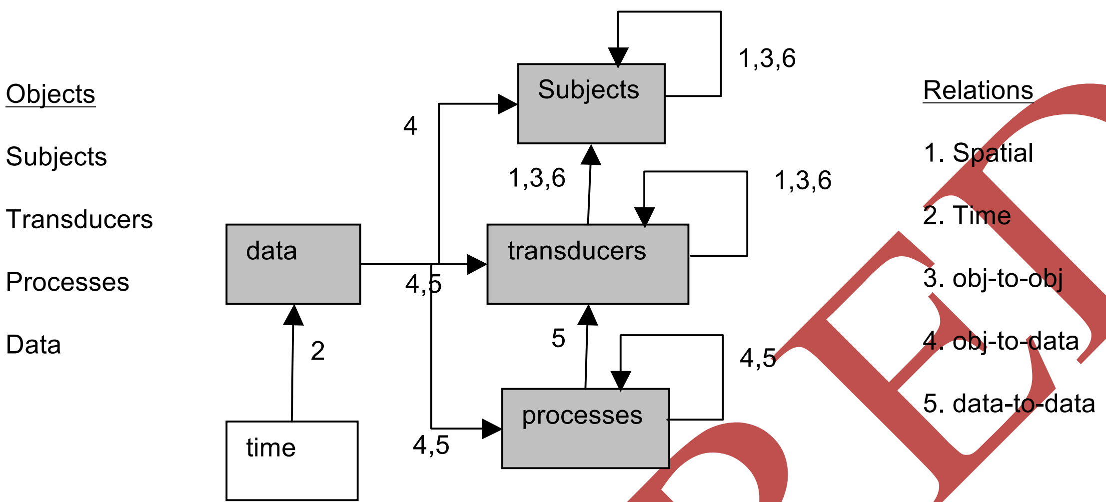


Figure 35 - Objects and Relationships

7.25.1.1 Position

The position relation identifies the location and attitude of a transducer relative to a spatial reference frame. The transducer may be positioned relative to another transducer or it may be positioned relative to a spatial datum such as ECEF. The position of a transducer may be a time dependent position. If so, then the position will be the result of a sensor measurement.

7.25.1.2 Absolute time

The time tag in the cluster provide relative timing of the objects within the cluster. To give then an absolute time the relative clock must be aligned with an absolute time reference. (i.e. UTC time). It must be made explicit that the time of a measurement coincides with the absolute time captured. Times may be captured that do no represent trigger time. This relationship identifies which time clock.

7.25.1.3 Object to Object Relation

This relation describes subjects to subjects, and transducer to object relations.

- Subject to Subject:
  - Relationship description
  - Direct-indirect object

  - Subject to Subject example:
    - In the statement, "The pH of water contained in the tank", water is the direct object and tank is the indirect object. "Contained in" is the relation.
- Transducer to Object:
  - Type of Object (Transducer or Subject)
  - Transducer to Object example (i.e. Dangle):
    - In the statement, "The pH of the water contained in the tank", The subject is the water, and the Transducer would be a pH sensor. This is the same as the previous dangle relationship. The dangled transducer measures or controls the property of the object. The property of the object in this case is pH.

7.25.1.4 Object to Data Relation

This relation associates transducers (object) to bindUids (data) and subjects to dataUnits.

- Transducer to BindUid Relation Example:
  - A rotational encoder measures the omega angular rotation of the gimbol stage relative to the gimbol base. The positional relation is not constant because the gimbol moves, therefore a bindUid is used to link in dynamic time stamped position.
  - The omega angular rotation of the position relationship between the gimbol stage and the base is described by linking to a sensor which measures its angular orientation.
  - The rotational actuator positions the omega angle of the gimbol stage relative to the gimbol base.
  - The omega angular rotation of the position relationship between the gimbol stage and the base is controlled by a rotational actuator.
- Subject to Data Example:
  - dataUnit1 (data) of dataSet B is associated the exhaust manifold (object) temperature,
  - dataUnit2 (data) of dataSet B is associated with the carburetor (object) vacuum pressure
  - Note: same dataSet, different objects, cant use subject to transducer relation.
- Subject to Data Example (Associate transducer data to a remote object using a target recognition process. more complex):
  - The subject identified in row 120 and column 434 of the data characteristic frame (data) appears to be a person (subject).
  - This can be a complex dynamic relation where several objects can be identified and associated with individual data units within a data steam. This may occur during or after data acquisition.

7.25.1.5 Data to Data Relation

This relation associates bindUIDs to processes and connects outputs to inputs.

- BindUID to Process. See bindUID discussion in section 7.23.3.3.2.
- Input to Output:
  - The output (source) of a transducer (transducerUID) connects to the input (sink) of a process (inputUID) e.g. RF receiver to a bandpass filter.
  - The output (source) of a process (outputUID) connects to the input (sink) of a transducer (transducerUID) e.g. wave form generator to an RF transmitter.

– The output (source) of a process (outputUID) connects to the input (sink) of a process (inputUID). e.g. mpeg 4 compression to a base64 encoding.

The input to Output example is the same as the previous Connect relationship. The connectivity or functional flow of data is described so that by the time data is received it may have gone through several processes. In order to understand what to do with the data we must know its processing heritage. For example, if the data has been compressed then Base64 encoded, then when you receive it you must first Base64 de-code then decompress in that order. To get back to the original data, the process the data went through as well as the order of the process must be understood. The connect relationship enables transducers and process inputs and outputs to be connected together to describe the data flow through the system.

8 TML Structure (Normative)

TML provides a self-contained lightweight XML envelope for efficient transport of transducer data as well as all necessary information to decode, process, analyze and understand the transducer data. Transducer system, spatial and temporal metadata are also characterized and carried along with the data, allowing analysis of transducer data and processing of the data only using information carried in the TML data stream.

TML achieves interoperability through several key features. These include:

- ability to plug TML behavior and response models into a standardized transducer definition
- ability to interface with and utilize arbitrary sensor output
- normalization of discordant datasets using TML behavior models

TML is composed of overall system description and transducer data from or to that system. Figure 36 below illustrates the overall structure of TML, including the system description and data. The system description includes summary descriptions of a transducer, process, relation, and cluster encoding. Note that this is not necessarily a complete TML model, and is intended only for illustrative purposes to show the overall structure of a TML description.


```
<?xml version="1.0" encoding="UTF-8"?>
<tml>
  <system name="SYS01">
    <sysClk>
      <period>
        <values>0.001</values>
      </period>
    </sysClk>
    <transducers>
      <transducer name="SYS01_CAMERA">
      <logicalDataModel>
        <dataSet id="CAMERA_IMAGE"></dataSet>
      </logicalDataModel>
      </transducer>
    </transducers>
    <process name=”PROCESS_1” uid=”SYS01_PCS”>
      <input>
        <procDataSet name="CAMERA_IMAGE"></dataSet>
      </input>
    </process>
    <relations>
      <posRelation UidRef="SYS01_CAMERA" refSystem="SYS01_GPS"/>
        <spatialCoord coordName="latitude">
          <values bindUid="SYS01_GPS_LAT"/>
        </spatialCoord>
      </posRelation>
    </relations>
    <clusterDesc name="SYS01_GPS_LLA">
      <dataUnitEncoding>
      </ dataUnitEncoding >
    </clusterDesc>
  </system>
<!—start data stream-->
  <data clk="263084829" ref="SYS01_TIMESTAMP">2006-03-02T14:39:41.04Z</data>
  <data clk="263085859" ref="SYS01_GPS_LLA">0.577041 -2.035342 0.000000</data>
  <data clk="12344" ref="SYS01_CAMERA_RGB24_BASE64">ABabdks2836875........ ARBKDK==</data>
<!—continue data stream-->
</tml>
```

Figure 36 - TML Static Data and Data Stream

Since TML may be considered data centric we will first examine TML data then we will discuss the various element for describing the data.. Below is a sample data tag as well as examples.

```
<data dateTime="2006-02-01T14:41:31.01Z"
reference="FULLYQUALIFIED_URI_SYS01_TEMP">223</data>
```

Example 1:

This example shows a data cluster at system clock time ‘12344’ from a transducer with transducer id ‘SYS01_TEMPERATURE_ASCII’.

```
<data clk="12344" ref="SYS01_TEMPERATURE_ASCII">223</data>
```

Example 2:

This example shows a data cluster at system clock time ‘88734’ from a transducer with transducer id ‘SYS01_CAMERA_RGB24_BASE64’.

```
<data clk="12344" ref="SYS01_CAMERA_RGB24_BASE64">ABadk26175....ARBKDK==</data>
```

Example 3:

```
<data clk='28118792' ref='SYS01_TEMPERATURE'>21.1</data>
<data clk='28118795' ref='SYS01_TIMESTAMP'>2005-08-26T16:31:49Z</data>
<data clk='28118800' ref='CAM'>AB12...base64 RGB24 image ...12Aa=</data>
<data clk='28118874' ref='SYS01_WIND'>9.8 3.1415</data>
```

8.1 TransducerML Schema Description

The following documentation was created with the assistance of XMLSpy.

schema location:	**\tml v0r6.xsd**
attribute form default:	**unqualified**
element form default:	**qualified**
targetNamespace:	**http://www.opengis.net/tml**

TML imports elements from the industry security manual namespace for security attributes. The security attributes are described in section 8.1.2

schema location:	**\IC-ISM-v2.xsd**
attribute form default:	
element form default:	**qualified**
targetNamespace:	**urn:us:gov:ic:ism:v2**

8.1.1. TML Elements

The root elements is of the schema are tml, system, transducer, or process. The elements are arranged in alphabetical order in three groups. First are the tml global elements, then the complex types and finally the attribute groups.

The following is a list of element, complex types and attributes groups described in this section.

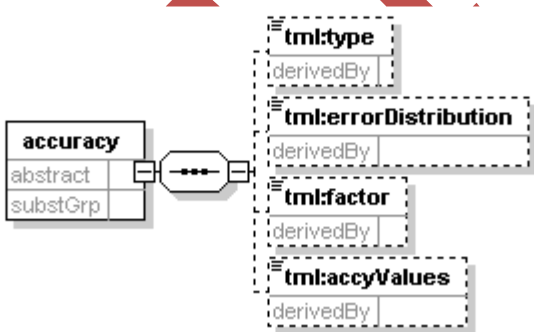
Elements	Complex types	Attr. groups
accuracy	**BindType**	**uid_uidRef**
cfSubSampling	**DataArrayType**	
clusterDesc	**IdentificationType**	
data	**ProcessType**	
dataUnit	**SpatialCoordType**	
logicalDataStructure	**SystemType**	
objToDataRelation	**TransducerType**	
objToObjRelation	**ValueType**	
process		

8.1.1.1 TML Global Elements

The following elements are described in this section in the order shown

8.1.1.1.1 element accuracy

diagram



namespace	http://www.opengis.net/tml	
properties	content	complex
children	tml:type tml:errorDistribution tml:factor tml:accyValues	
used by	element	clusterDesc/timeTag
	complexType	ValueType
annotation	documentation	accuracy is in terms of the data value before adjustment by mult and offset. if a characteristic frame (i.e. number of values) of values of accuracy, then each value corresponds to the corresponding Characteristic Frame position or interval

8.1.1.1.1.1 element accuracy/type

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default absolute

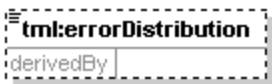
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values: relative, absolute, systematic, random. default is absolute

8.1.1.1.1.2 element accuracy/errorDistribution

diagram



namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default gaussian

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed Values: gaussian, chi, chi2, possion, gamma. default is gaussian

8.1.1.1.1.3 element accuracy/factor

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default 1sigma

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation allowed values: 1sigma, 2sigma, 3sigma, 4sigma, 5sigma, 6sigma, RMS, RSS, percent, range. Default is 1sigma

8.1.1.1.1.4 element accuracy/accyValues

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

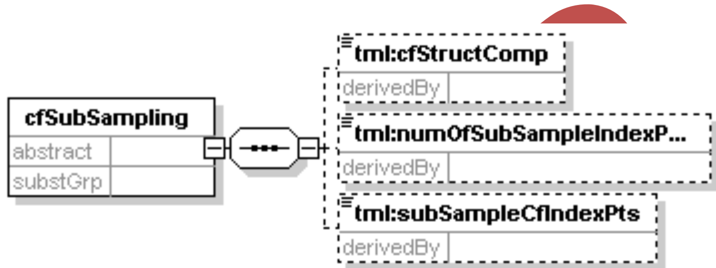
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation A single accyValue relates to whole range of parent coordinates (e.g. data or prop). If accyValue is variable over the parent coordinates then there shall be a one-to-one correspondence between the accyValues and the parent coordinates. Use mult and offset to describe variances over CF

8.1.1.1.2 element cfSubSampling

diagram



namespace http://www.opengis.net/tml
properties content complex
children **tml:cfStructComp** **tml:numOfSubSampleIndexPoints** **tml:subSampleCfIndexPts**
used by elements **responseModels** **TransducerType/temporalModel/ambiguityTime** **objToDataRelation/object** **TransducerType/spatialModel** **TransducerType/temporalModel/tcfOffsetTime**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation the CFSubSampling can be used for chipping part of a large dataArray out and for reducing the number of points within an array for which to associate modelling parameters. there is one subsampling element set of points for each component structure (col, row, plane). index numbers of col, row or plane position within the CFs are listed for which corresponding modelling points will be associated. sample points are separated by commas, ranges are indicated by ... between numbers which indicates a continuous interval for a single sample. interpolation between samples uses logical structure.

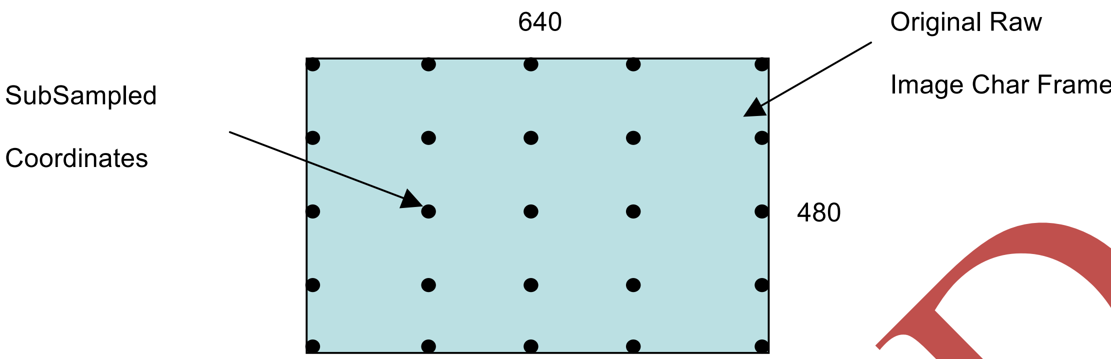


Figure 37 - Sub-Sampling Example

```
Below is the TML code for the example above:
<cfSubSampling
  <cfStructComp>rows</cfStructComp>
  <numbOfSubSampleIndexPoints>5 </numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>0, 119, 239, 359, 479 </subSampleCfIndexPts>
</cfSubSampling>

<cfSubSampling
  <cfStructComp>columns</cfStructComp>
  <numbOfSubSampleIndexPoints>5 </numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>0, 159, 319, 479, 659</subSampleCfIndexPts>
</cfSubSampling>
```

The two sub-sampling examples below produce an identical result.

Example 1:

```
<cfSubSampling>
<cfStructComp>rows</cfStructComp>
  <numbOfSubSampleIndexPoints>5</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>0, 119, 239, 359, 479 </subSampleCfIndexPts>
</cfSubSampling>
<cfSubSampling>
  <cfStructComp>columns</cfStructComp>
  <numbOfSubSampleIndexPoints>5</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>0, 159, 319, 479, 659</subSampleCfIndexPts>
</cfSubSampling>
```

Example 2:

```
<cfSubSampling
  <cfStructComp>rows</cfStructComp>
  <numbOfSubSampleIndexPoints>5</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>0…479</subSampleCfIndexPts>
</cfSubSampling>
<cfSubSampling
  <cfStructComp>columns</cfStructComp>
  <numbOfSubSampleIndexPoints>5</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>0…659</subSampleCfIndexPts>
</cfSubSampling>
```

The next example shows how a chip of an image may be described.

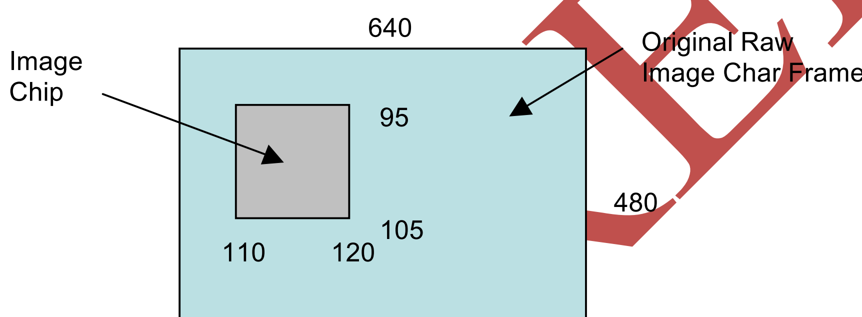


Figure 38 - Sub-Sampling Example

The following subsampling chips out a section of data:

```
<cfSubSampling>
  <cfStructComp>rows</cfStructComp>
  <numbOfSubSampleIndexPoints>10</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>95…105</subSampleCfIndexPts>
</cfSubSampling>
<cfSubSampling>
  <cfStructComp>columns</cfStructComp>
  <numbOfSubSampleIndexPoints>10</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>110…120</subSampleCfIndexPts>
</cfSubSampling>
```

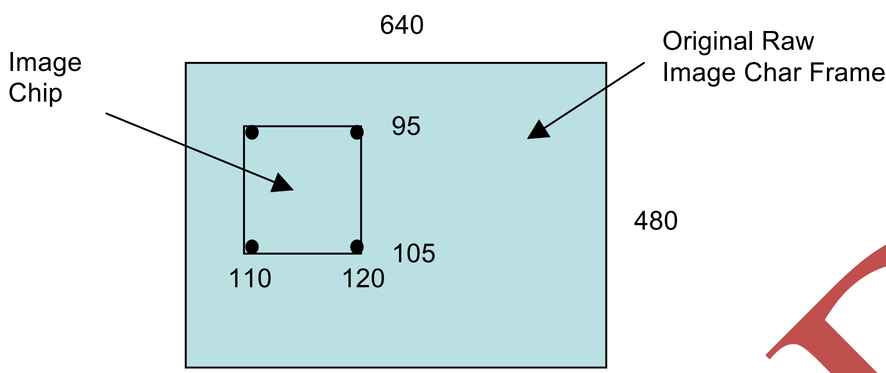



Figure 39 - Sub-Sampling Example

The following sub-sampling example takes out only corner points of the image chip:

```
<cfSubSampling>
  <cfStructComp>rows</cfStructComp>
  <numbOfSubSampleIndexPoints>2</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>95,105</subSampleCfIndexPts>
</cfSubSampling>
<cfSubSampling>
  <cfStructComp>columns</cfStructComp>
  <numbOfSubSampleIndexPoints>2</numbOfSubSampleIndexPoints>
  <subSampleCfIndexPts>110,120</subSampleCfIndexPts>
</cfSubSampling>
```

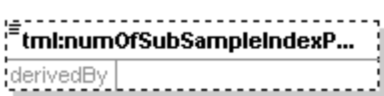
8.1.1.1.2.1 element cfSubSampling/cfStructComp

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
	default	column					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	Allowed values: column, row, plane. default is column. One cfSubSampling element for each cfStructComp required.					

8.1.1.1.2.2 element cfSubSampling/numOfSubSampleIndexPoints

diagram



namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: Allowed values: positive integers from 1 to the number of columns, rows, or planes in the data structure. This number indicates the number of samples in the cfSubSampleIndexPts.

8.1.1.1.2.3 element cfSubSampling/subSampleCfIndexPts

diagram

namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex

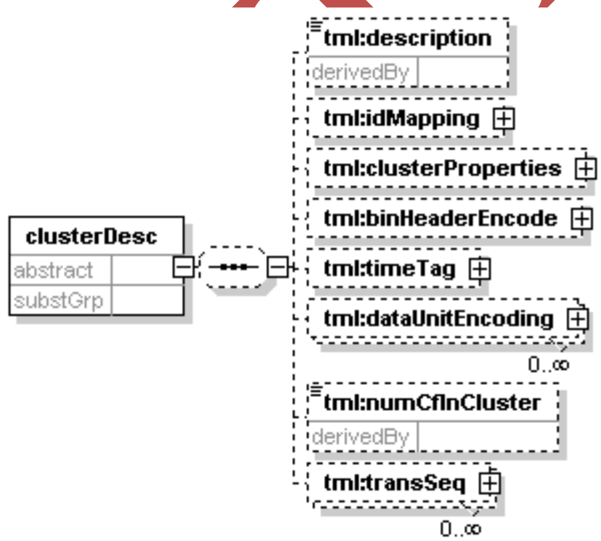
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: use same rules as points under value

8.1.1.1.3 element clusterDesc

diagram



namespace	http://www.opengis.net/tml
properties	content complex
children	**tml:description tml:idMapping tml:clusterProperties tml:binHeaderEncode tml:timeTag tml:dataUnitEncoding tml:numCfInCluster tml:transSeq**
used by	element **SystemType/clusterDescriptions**

attributes

Name	Type	Use	Default	Fixed	Annotation

	uid_uidRef Attribute Group	optiona			documentation	See uid_uidRef Attribute Group
	ismSecurityOptionsGroup	optional			documentation	See ismSecurityOptionsAttribute Group

annotation	documentation	The clusterDesc element describes the explicit structure of raw data within a TML cluster. The clusterDesc represents a mapping from the logical data model to actual data and provides the encoding/decoding information to enable the understanding of the transducer data. The Data Cluster Description describes and provides the implicit structure of the data message that is transmitted in a data cluster. The Data Cluster Description has an identifier and the encoding element that correlates to a data model and specifies the implicit structure of the data message. The specifics of how the implicit structure is encoded is in two parts, the first part is the use of the CF, set and unit layout, and the second part is the encoding details. The encoding details specify the information necessary to split the data out into the corresponding data fields as specified in the Data Model. An empty clusterDesc tag in a data stream indicates that this cluster is no longer contained in the data stream.

Issue Name: tml rwg 1.2 and 1.6 (M. Botts, 3 Oct 2006)

Issue Description: Recommend that TML derive from or use the SWE Common data definition schemas (swe:parameters and swe:data) for defining tml:clusterDescr andtml:LogicalDataModel. These changes would be relatively simple with minimal disruption of the rest of TML, but would provide significant benefits in allowing better coordination of TML with other SWE framework components. SWECommon parameters and data definitions are utilized throughout the SWEframework for defining inputs, outputs, parameters, and properties in SensorML, for defining data components and encodings in Common Observation (the primary observation definition derived from O&M), for defining tasking parameters in SPS, and for defining ObservationOffering in SOS

Resolution: Identify as an issue and document as future harmonization work with SensorML (S. Havens, 3 Oct 2006)

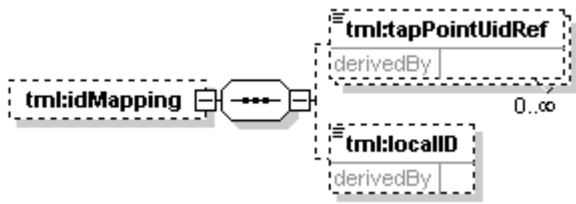
8.1.1.1.3.1 element clusterDesc/description

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	description of the data cluster					

8.1.1.1.3.2 element clusterDesc/idMapping

diagram



namespace http://www.opengis.net/tml

properties
isRef 0
minOcc 0
maxOcc 1
content complex

children **tml:tapPointUidRef** **tml:localID**

8.1.1.1.3.3 element clusterDesc/idMapping/tapPointUidRef

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc unbounded
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation dataUidRef of the tap point in the system to which this cluster corresponds. UID of the transducer, process input process output, or connection node from which or to which this cluster relates. This is the UID used in the data header (i.e. reference attribute in data start tag). Is some cases a data in a single cluster may come from multiple dataUid tap points.

8.1.1.1.3.4 element clusterDesc/idMapping/localID

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex

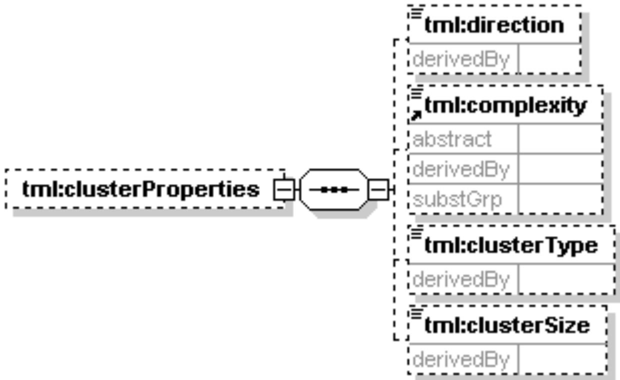
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation short ID used in the data header (i.e. ref attribute in data start tag)

8.1.1.1.3.5 element clusterDesc/clusterProperties

diagram



namespace http://www.opengis.net/tml

properties
- isRef 0
- minOcc 0
- maxOcc 1
- content complex

children **tml:direction** **tml:complexity** **tml:clusterType** **tml:clusterSize**

8.1.1.1.3.6 element clusterDesc/clusterProperties/direction

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
- isRef 0
- minOcc 0
- maxOcc 1
- content complex
- default fromSystem

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation The direction identifies the cluster as coming from a sensor or a receiver or a cluster going to an actuator or a transmitter. This element is enumerated with fromSystem and toSystem. Allowed Values: fromSystem, toSystem. default fromSystem

8.1.1.1.3.7 element clusterDesc/clusterProperties/complexity

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
- isRef 0
- minOcc 0
- maxOcc 1
- content complex
- default 1A

attributes

Name	Type	Use	Default	Fixed	Annotation

	bindUid	**xs:string**	optional	documentation	See BindType
	bindUidRef	**xs:string**	optional	documentation	See BindType

annotation documentation The complexity element gives the receiving system some indication of the resources required to handle, and process the data cluster into the logical model. The following table is a strawman template of what complexity may entail. The complexity element is enumerated with 1A through 5E.. Allowed Values: 1A - 1F, 2A -2F, 3A - 3F, 4A - 4F, 5A - 5F. default 1A

Table 2 – Notional Complexity Levels based on Required Resources

	Memory Requirements	<100 MB	100 MB - 1 GB	1 GB - 10 GB	10 GB - 100 GB	>100GB
	Real-time FLOPS	<10 MFLOPS	10 MFLOPS - 100MFLOPS	100 MFLOPS - 10 GFLOPS	10 GFLOPS - 1 TFLOPS	> 1 TFLOPS
File Size	Realtime Bandwidth					
<100KB	<100Kb/s	1A	1B	1C	1D	1E
100KB - 1MB	100Kb/s - 1Mb/s	2A	2B	2C	2D	2E
1MB - 10MB	1Mb/s - 10Mb/s	3A	3B	3C	3D	3E
10MB - 1GB	10 - 100Mb/s	4A	4B	4C	4D	4E
>1GB	>100Mb/s	5A	5B	5C	5D	5E

FLOPS: FLoating point Operations Per Second

8.1.1.1.3.8 element clusterDesc/clusterProperties/clusterType

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default binary

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation The cluster type identifies the data contents as either XML or binary. Allowed values: binary, packedXML, verboseXML. Default: binary

8.1.1.1.3.9 element clusterDesc/clusterProperties/clusterSize

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

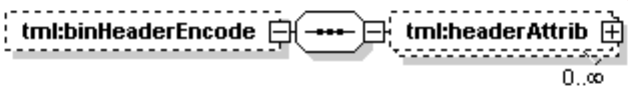
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Integer number of bytes in Cluster

8.1.1.1.3.10 element clusterDesc/binHeaderEncode

diagram



namespace http://www.opengis.net/tml

properties

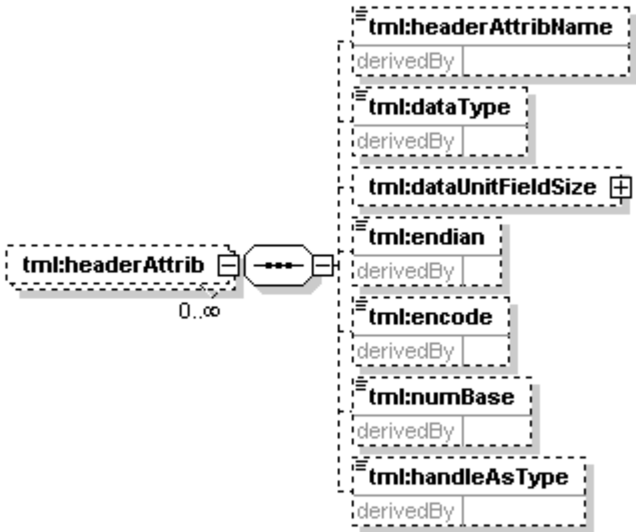
isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:headerAttrib**

annotation documentation This is the binary header encoding. If the data cluster type is binary, then this element is needed in order to describe the name, order and encodings of the fields in the header. If cluster type is binary this field describes the encoding of the header attributes. binary files will contain only the contents of the attributes and not the attribute tag. The binary header will not contain the left carrot and the letters "data" at the beginning of the header either, nor the right carrot at the end of the header. The attribute values will be packed without the attribute tag identifier.

8.1.1.1.3.11 element clusterDesc/binHeaderEncode/headerAttrib

diagram



namespace http://www.opengis.net/tml

properties
isRef 0
minOcc 0
maxOcc unbounded
content complex

children **tml:headerAttribName tml:dataType tml:dataUnitFieldSize tml:endian tml:encode tml:numBase tml:handleAsType**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

annotation documentation ref, reference, dateTime, contents and ismClass attributes will be encoded and handled as "string" type

8.1.1.1.3.12 element clusterDesc/binHeaderEncode/headerAttrib/headerAttribName

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default ref

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation Allowed values: ref, clk, reference, dateTime, contents, seq, total, ismClass. Default ref

8.1.1.1.3.13 element clusterDesc/binHeaderEncode/headerAttrib/dataType

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	number

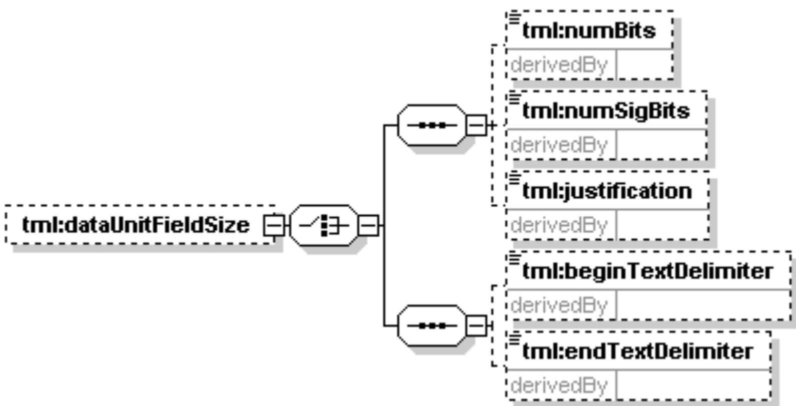
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values: text, number. Default is number.

8.1.1.1.3.14 element clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize

diagram



namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:numBits tml:numSigBits tml:justification tml:beginTextDelimiter tml:endTextDelimiter**	

8.1.1.1.3.15 element clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize/numBits

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	8

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation	documentation	number of bits. This element is needed if the field size is fixed. For example, if 6 bit camera pixels are being transferred as 8 bit bytes, then 8 is the numBits and 6 is the numSigBits. default 8

8.1.1.1.3.16 element clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize/numSigBits

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties

isRef	0
minOcc	0
maxOcc	1
content	complex
default	8

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation number of significant bits. This element is needed if the field size is fixed. For example, if 6 bit camera pixels are being transferred as 8 bit bytes, then 8 is the numBits and 6 is the numSigBits. default 8

8.1.1.1.3.17 element clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize/justification

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties

isRef	0
minOcc	0
maxOcc	1
content	complex
default	right

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation if numSigBits is less than numBits this element indicates how the significant bits are justified. Allowed values: left, right. Default: right

8.1.1.1.3.18 element clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize/beginTextDelimiter

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties

isRef	0
minOcc	0
maxOcc	1

content complex

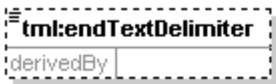
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation This element describes the pattern of the bytes which delimits the data The delimiter used to separate variable size dataUnits in cluster when encode is text (utf or ucs). default delimiter is none. empty tag means none.

8.1.1.1.3.19 element clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize/endTextDelimiter

diagram



namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation delimiter used to separate variable size dataUnits in cluster when encode is text (utf or ucs). default delimiter is none. Empty tag means none

8.1.1.1.3.20 element clusterDesc/binHeaderEncode/headerAttrib/endian

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default little

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation If the endian is big, then the order is most significant bit to least significant bit. If the endian is little, then it is the opposite. Allowed values: big, little. default little

8.1.1.1.3.21 element clusterDesc/binHeaderEncode/headerAttrib/encode

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1

content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation This element describes how the binary data is encoded. The following tables illustrate the various possibilities for use in encoding both ASCII and binary data within TML. Allowed values: ucs16, utf8, signInt, unsignInt, real, bcd. default unsignInt.

Table 3 - Text Encoding data types (used for any clusterType)

Encoding data type	Explanation
utf8	8 bits per character.
ucs16	16 bit per character

Table 4 - Number Encoding data types (only used for clusterType of binary)

Encoding data type	Explanation
signedInt	Field width specified by numSigBits 2's complement signed integer
unsignInt	Field width specified by numSigBits binary unsigned number
real	Standard IEEE 754 32 or 64 bit floating point number 16 or 32 bit indicated by numSigBits
bcd	4 bits per number – binary coded decimal Field width indicated by numSigBits, number of digits equal to numSigBit/4
utf8	Each number is represented as an 8 bit text number 0-9 with the following characters allowed: "+", "-", "E" Field width indicated by numSigBits, number of characters equal to numSigBit/8
ucs16	Each number is represented as an 16 bit text number 0-9 with the following characters allowed: "+", "-", "E" Field width indicated by numSigBits, number of characters equal to numSigBit/16

8.1.1.1.3.22 element clusterDesc/binHeaderEncode/headerAttrib/numBase

diagram

namespace http://www.opengis.net/tml

type [tml:BindType]

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default 10

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation This is the number base of the data When numbers are encoded as text the number base must be understood. Allowed values: 2, 8, 10, 16, 32, 64, 128. default 10

8.1.1.1.3.23 element clusterDesc/binHeaderEncode/headerAttrib/handleAsType

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

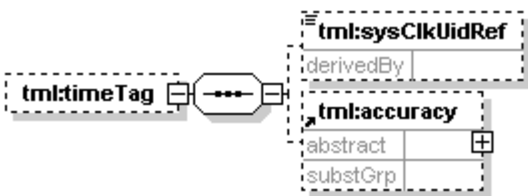
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: This element describes how the text or number value be handled in the client application. Allowed values: anuURI, Boolean, byte, double, float, short, string, int, integer, long, nonNegativeInteger, nonPositiveInteger, positiveInteger, unsignedByte, unsignedInt, unsignedShort, unsignedLong.

8.1.1.1.3.24 element clusterDesc/timeTag

diagram



namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:sysClkUidRef** **tml:accuracy**	

annotation — documentation: This element describes what time tag is used for this cluster. Useful when parent systems normalize clocks from child components. This element also describes how accurately the sysClk value is applied to the cluster start instant. This is different from the accuracy of the system clock.

8.1.1.1.3.25 element clusterDesc/timeTag/sysClkUidRef

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

attributes

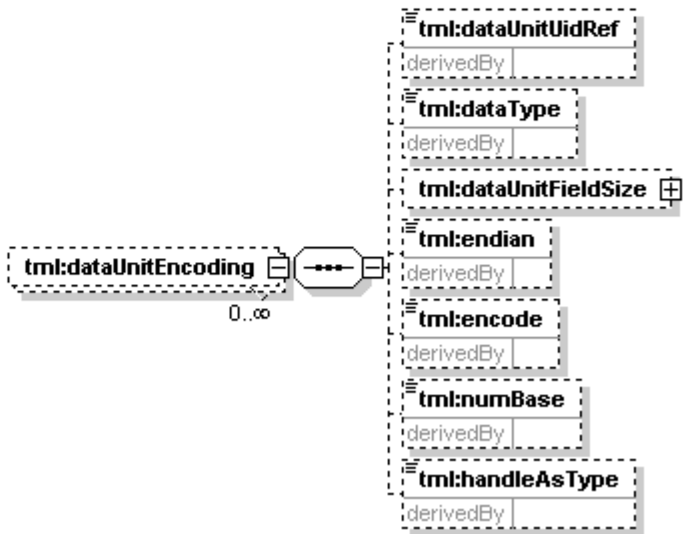
Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType

Name	Type	Use	Default	Fixed	Annotation
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: if clk is used in the start tag and multiple clocks are used in a system. Default is the first parent system clock

8.1.1.1.3.26 element clusterDesc/dataUnitEncoding

diagram



namespace: http://www.opengis.net/tml

properties:
- isRef 0
- minOcc 0
- maxOcc unbounded
- content complex

children: **tml:dataUnitUidRef tml:dataType tml:dataUnitFieldSize tml:endian tml:encode tml:numBase tml:handleAsType**

attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation — documentation: This element specifies how the data value from or to the transducer is represented in digital form. This element describes the encoding of the dataUnit identified in the dataUnitUidRef child element. The dataUnits within a data cluster are exchanged as printable text when the clusterType element is XML. Numeric characters may be interpreted as a text character or as a numeric character depending on the state of the dataType field (encodedBinaryBlob, text or number). Some clusters which represent only an event from a source or a trigger are empty and may not contain any dataUnits.

8.1.1.1.3.27 element clusterDesc/dataUnitEncoding/dataUnitUidRef

diagram

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties:
- isRef 0
- minOcc 0
- maxOcc 1
- content complex

attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType

	Name	Type	Use	Default	Fixed	Annotation	
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation UID of the dataUnit from the logical structure.

8.1.1.1.3.28 element clusterDesc/dataUnitEncoding/dataType

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default text

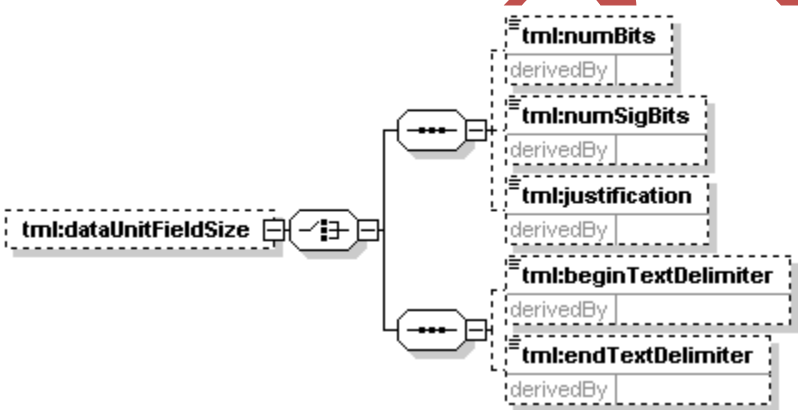
attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation Allowed values: text, number, binBlob. Default is text.

8.1.1.1.3.29 element clusterDesc/dataUnitEncoding/dataUnitFieldSize

diagram



namespace http://www.opengis.net/tml
properties
isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:numBits** **tml:numSigBits** **tml:justification** **tml:beginTextDelimiter** **tml:endTextDelimiter**

8.1.1.1.3.30 element clusterDesc/dataUnitEncoding/dataUnitFieldSize/numBits

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default 8

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation default 8

8.1.1.1.3.31 element clusterDesc/dataUnitEncoding/dataUnitFieldSize/numSigBits

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default 8

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation Default 8

8.1.1.1.3.32 element clusterDesc/dataUnitEncoding/dataUnitFieldSize/justification

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default right

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation if numSigBits is less than numBits this element indicates how sigbit are justified. Allowed values: left, right. Default: right

8.1.1.1.3.33 element clusterDesc/dataUnitEncoding/dataUnitFieldSize/beginTextDelimiter

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

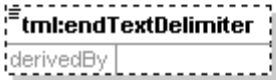
attributes	Name	Type	Use	Default	Fixed	Annotation

	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation delimiter used to separate variable size dataUnits in cluster when encode is text (utf or ucs). default delimiter is none. Empty tag means none.

8.1.1.1.3.34 element clusterDesc/dataUnitEncoding/dataUnitFieldSize/endTextDelimiter

diagram



namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation default delimiter is none. Empty tag means none.

8.1.1.1.3.35 element clusterDesc/dataUnitEncoding/endian

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default little

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation Allowed values: big, little. default little

8.1.1.1.3.36 element clusterDesc/dataUnitEncoding/encode

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation Allowed values: ucs16, utf8, signInt, unsignInt, real, complex, bcd. default utf8. When clusterType is "XML" only utf8 is allowed in cluster. All types are allowed when clusterType is binary. Complex values are exchanged as two phenomenon (mag and phase or real and imaginary components) or as a single complex number.

8.1.1.1.3.37 element clusterDesc/dataUnitEncoding/numBase

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
	default	10					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	when numbers are encoded as text the number base must be understood. Allowed values: 2, 8, 10, 16, 32, 64, 128. default 10					

8.1.1.1.3.38 element clusterDesc/dataUnitEncoding/handleAsType

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	This element describes how the text or number should be handled in the client application. Allowed values: anyURI, Boolean, byte, double, float, short, string, int, integer, long, nonNegativeInteger, nonPositiveInteger, positiveInteger, unsignedByte, unsignedInt, unsignedShort, unsignedLong.					

8.1.1.1.3.39 element clusterDesc/numCfInCluster

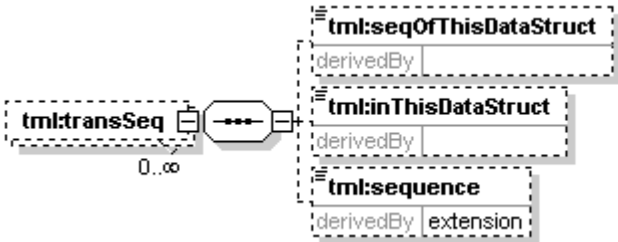
diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation	documentation	number of characteristic frames in a cluster or the number of clusters which comprise a large characteristic frame. default = 1. example: 2 means 2 CF per cluster, -2 means 2 clusters per CF. Allowed values: signed integer. zero not allowed.

8.1.1.1.3.40 element clusterDesc/transSeq

diagram



namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
children	**tml:seqOfThisDataStruct tml:inThisDataStruct tml:sequence**	
annotation	documentation	This is the order in which data is sent in the cluster or CF (whichever is larger) relative to the logical data structure. The order of structure components are listed from lowest freq to highest frequency order. If transport sequence is blank then the sequence is the same as the logical order (sequence) for that structure component.

8.1.1.1.3.41 element clusterDesc/transSeq/seqOfThisDataStruct

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
	default	

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation	documentation	Sequence of (in this element - seqOfThisDataStruct) in the data structure identified in the next element (inThisDataStruct). seqOfBitsInUnit, seqOfUnitsInSets, seqOfSetsInCf, seqOfCfInClust. Identify the dataStructComponent in this element by dataUidRef. dataUid of the cluster is "cluster"

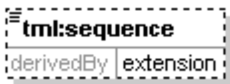
8.1.1.1.3.42 element clusterDesc/transSeq/inThisDataStruct

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	unbounded					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	Sequence of the data structure components identified in the previous element (seqOfThisDataStruct) in the data structure identified in this element (inThisDataStruct). seqOfBitsInUnit, seqOfUnitsInSets, seqOfSetsInCf, seqOfCfInClust. Identifty the dataStructComponent in this element by dataUidRef. dataUid of the cluster is "cluster"					

8.1.1.1.3.43 element clusterDesc/transSeq/sequence

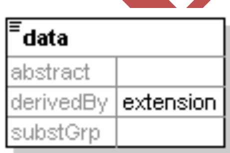
diagram



namespace	http://www.opengis.net/tml						
type	extension of **tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	see BindType
	bindUidRef	**xs:string**	optional			documentation	see BindType
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group
annotation	documentation	Allowed values; The sequence shall contain a string of value separated by a comma. Each value can be a positive integer or a range. ranges shall be indicated by two integer numbers separated by three sequential decimal points (....) to indicate a run from the first number to the second					

8.1.1.1.4 element data

diagram



namespace	http://www.opengis.net/tml						
type	extension of **xs:string**						
properties	content	complex					
used by	element	**tml**					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	ref	**xs:string**	optional			documentation	alias or short id reference of transducer or process producing this data
	clk	**xs:integer**	optional			documentation	sys clock state at trigger point to data cluster. For low sampling frequency transducers this high frequency clock state may not be required. A full

					dateTime attribute may suffice for time synchronization of data.
reference	**xs:anyURI**	optional		documentation	this is the full UID reference to the cluster description
dateTime	**xs:dateTime**	optional		documentation	Full qualified date and time of transducer or process producing this data. For low sampling frequency transducers this high frequency clock state may not be required. A full dateTime attribute may suffice for time synchronization of data.
contents	**derived by: xs:string**	optional	exp	documentation	If a binary stream header does not contain a contents field then the data cluster is by default explicit data. This field is encoded as a binary (2-bits) "00" in a binary file if the field is contained.
seq	**xs:integer**	optional		documentation	if no "total" attribute exist then this attribute can be used to number the data elements like a count, this enables the receipt end to determine if any data clusters were lost.
total	**xs:integer**	optional		documentation	total in sequence e.g. 1 of 4, 2 of 4. 1 and 2 being the seq number and 4 being the total
ismclass	**ism:ClassificationType**			documentation	security classification of each data cluster. Overall data classification of transducer data in clusterDescription. Overall classification of file or stream in tml start tag.

annotation — documentation

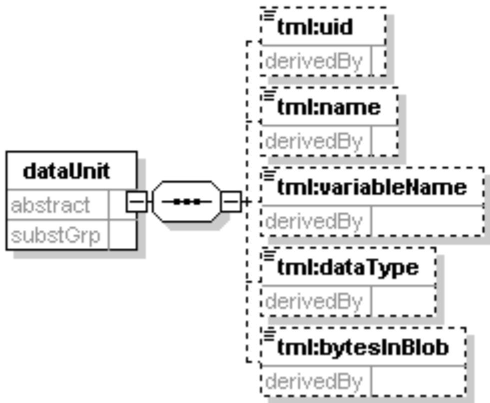
this element carries the date to or from transducer systems. The data element will carry a single instance or a continuous stream of a condition or set of synchronous conditions time tag to the precise instant of creation. There is no XML mark-up of data within the data tag. A system description will describe the decoding and understanding of the data within the data tag.

The tml:data element includes the attributes of ref, clk, dateTime, reference, contents, seq, and total. Ref is a short local reference to a Data Cluster Description. Clk is an integer number which is the value of a local stable counter associated with the enclosed data. DateTime is an optional attribute that can be used to associate a fully qualified date timestamp with a data cluster. The format is "CCYY-MM-DDThh:mm:ss.sssZ". Reference is a fully qualified URI to the Data Cluster Description, System and Transducer. If a binary stream header does not contain a contents field then the data cluster is by default explicit data. Contents is enumerated as imp or exp (implicit or explicit). This field is encoded as a binary (2-bits) "00" in a binary file if this field is contained. If no "total" attribute exist then the "seq" attribute can be used to number the data elements like a count, this enables the receipt end to determine if any data clusters were lost. Total is the total in the sequence (e.g. 1 of 4 or 2 of 4, where 1 and 2 are the seq number and 4 is the total).

The time tag is particularly important to TML, as it allows precise determination of when a transducer measurement was taken. This time reference is put into the data cluster and represents the time at which the trigger occurred to either sample data (for receivers) or act on data (for transmitters). In the case of transmitters, the time tag may represent a time in the future. This is meant for the transducer system to act at a future time. If no time tag is present in data for a transmitter, then the action shall occur as soon as possible.

8.1.1.1.5 element dataUnit

diagram



namespace http://www.opengis.net/tml

properties content complex

children tml:uid tml:name tml:variableName tml:dataType tml:bytesInBlob

used by element DataArrayType/dataSet

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation an elemental unit of data. one description for each unit. A dataUnit element contains the description for the individual data value corresponding to an instance of a phenomenon. One or more data units are contained in a data set. As transducer data gets further from the transducer (i.e. processed), it may be necessary to handle the data unit as a digital blob where it no longer relates directly to a phenomenon. To understand how it indirectly relates it would be necessary to follow the process path to the phenomenon. This element includes the identification and dataType elements as well as the bitsInBlob element.

8.1.1.1.5.1 element dataUnit/uid

diagram

namespace http://www.opengis.net/tml

type tml:BindType

properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation uid of dataUnit

8.1.1.1.5.2 element dataUnit/name

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation name of dataUnit

8.1.1.1.5.3 element dataUnit/variableName

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation mathematical variable name used in the transformation equations. Index of component is the order in the sequence in the Logical Data Structure (LDS).

8.1.1.1.5.4 element dataUnit/dataType

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties isRef 0
minOcc 0
maxOcc 1
content complex
default number

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values: number, complexNumber, text, or binaryBlob. default is number

8.1.1.1.5.5 element dataUnit/bytesInBlob

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties isRef 0
minOcc 0

maxOcc 1
content complex

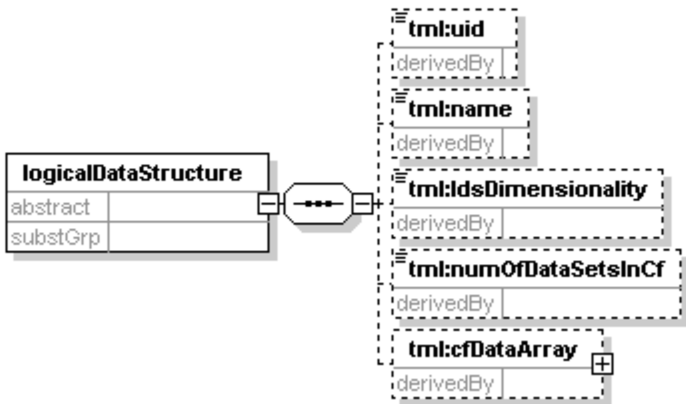
attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation If dataType is binaryBlob then number of bytes in the binary blob. Not used for transducer structures, only for process structures.

8.1.1.1.6 element logicalDataStructure

diagram



namespace http://www.opengis.net/tml

properties content complex

children **tml:uid** **tml:name** **tml:ldsDimensionality** **tml:numOfDataSetsInCf** **tml:cfDataArray**

used by elements **ProcessType/input** **ProcessType/output**
complexType **TransducerType**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

annotation documentation the logical structure of data (i.e. of the characteristic frame). The tcfDataArray element defines the logical layout of data. For example, data can be laid out in line scans, an array, or other ways. A tcfDataArray element is a homogeneous collection of dataSets. A data array is composed of one or more data sets. This is not necessarily the structure or order that data is communicated in. The transmission order is defined in the cluster description. The transmission order is defined relative to the logical order.

8.1.1.1.6.1 element logicalDataStructure/uid

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation uid of LDS

8.1.1.1.6.2 element logicalDataStructure/name

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

- isRef 0
- minOcc 0
- maxOcc 1
- content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation name of LDS

8.1.1.1.6.3 element logicalDataStructure/ldsDimensionality

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

- isRef 0
- minOcc 0
- maxOcc 1
- content complex
- default 0

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values: 0, 1, 2, 3. Default is 0. dimensionality of the logical data structure (LDS). number of structure components used for giving hints for data representation. 0 dim is a single value, 1 dim is a series of columns, rows or planes, 2 dim is any order of two structure components (col-row, col-plane, or row-plane), and a 3 dim is any order of three structure components col-row-plane

8.1.1.1.6.4 element logicalDataStructure/numOfDataSetsInCf

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

- isRef 0
- minOcc 0
- maxOcc 1
- content complex

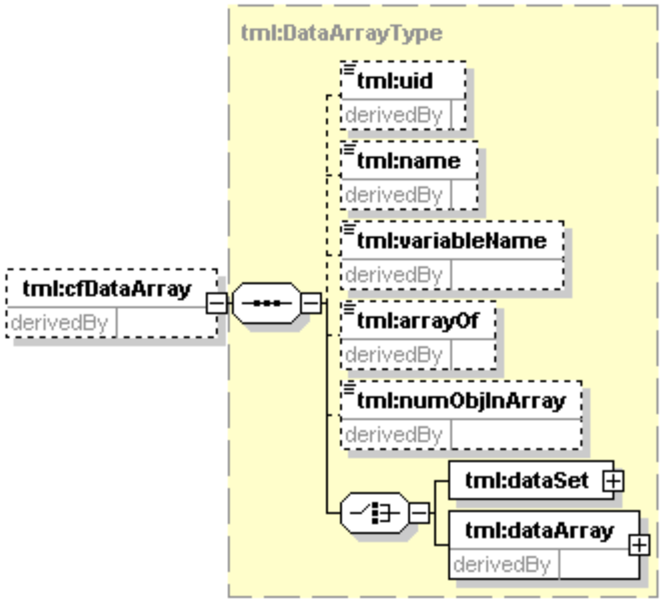
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Number of datasets or dataArrays in the Characteristic Frame. Allowed Value: positive integer. Default:1

8.1.1.1.6.5 element logicalDataStructure/cfDataArray

diagram



namespace http://www.opengis.net/tml
type **tml:DataArrayType**
properties isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:uid tml:name tml:variableName tml:arrayOf tml:numObjInArray tml:dataSet tml:dataArray**

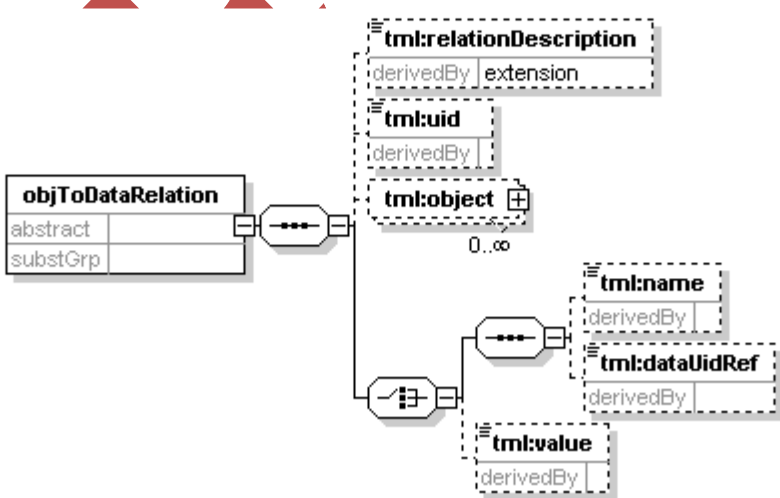
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation logical data structure of the characteristic frame. Lowest frequency array first.

8.1.1.1.7 element objToDataRelation

diagram



namespace http://www.opengis.net/tml
properties content complex

children: **tml:relationDescription** **tml:uid** **tml:object** **tml:name** **tml:dataUidRef** **tml:value**

used by: elements **tml/extSysRelations** **SystemType/relations**

attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation: See uid_uidRef Attribute Group
SecurityOptionsAttributeGroup		optional			documentation: See ismSecurityOptionsAttribute Group

annotation: documentation: Connects transducer to bindUids. Associate transducer data to a (remote) object. This may occur after data acquisition. An object is either a transducer, subject or their properties. Many subjects may be related to data in a dataArray. The objects can be related to data units, sets and arrays to subjects.

8.1.1.1.7.1 element objToDataRelation/relationDescription

diagram:

namespace: http://www.opengis.net/tml

type: extension of **tml:BindType**

properties:
- isRef: 0
- minOcc: 0
- maxOcc: 1
- content: complex

attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation: See BindType
bindUidRef	**xs:string**	optional			documentation: See BindType

annotation: documentation: description of the signal or the property relation

8.1.1.1.7.2 element objToDataRelation/uid

diagram:

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties:
- isRef: 0
- minOcc: 0
- maxOcc: 1
- content: complex

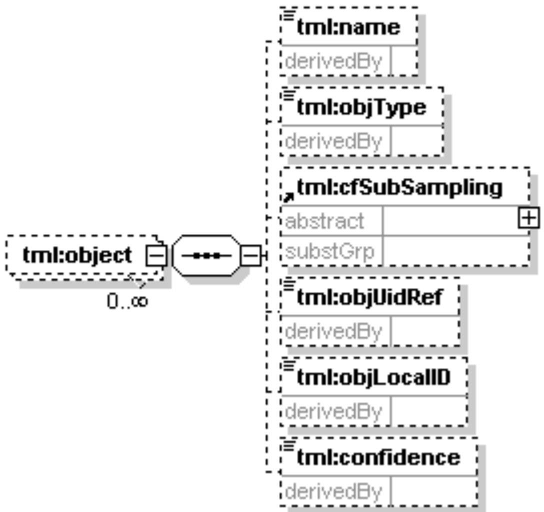
attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation: See BindType
bindUidRef	**xs:string**	optional			documentation: See BindType

annotation: documentation: connection or node UID of the connection signal data or property relationship

8.1.1.1.7.3 element objToDataRelation/object

diagram



namespace: http://www.opengis.net/tml

properties:

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children: **tml:name** **tml:objType** **tml:cfSubSampling** **tml:objUidRef** **tml:objLocalID** **tml:confidence**

attributes:

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

annotation: documentation — Object can be a single transducer (dangle relation), a single dataUID, or many subjects can be related to a single data unit. probabilities can be assigned to each relation.

8.1.1.1.7.4 element objToDataRelation/object/name

diagram

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties:

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes:

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation: documentation — name of the object

8.1.1.1.7.5 element objToDataRelation/object/objType

diagram

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default subject

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation identify object as a transducer or a subject. Allowed Values: subject, transducer. Default: subject

8.1.1.1.7.6 element objToDataRelation/object/objUidRef

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation UID of the object (subject or transducer, or probable subject). local id of the subject if multiple ids are used to associate with each cell of the logical structure.

8.1.1.1.7.7 element objToDataRelation/object/objLocalID

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation if localId assigned to objUidRef for building CF of obj to data (i.e.CF) relationships. Sequence of values is the same as the sequence in the data (logical data structure)

8.1.1.1.7.8 element objToDataRelation/object/confidence

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0

	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	confidence values match same sequence as logical data structure (if multiple values in data structure)					

8.1.1.1.7.9 element objToDataRelation/name

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

8.1.1.1.7.10 element objToDataRelation/dataUidRef

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	UID of the data reference. Archived data streams will have a UID indicative of the data source, time, and clk count of the start.					

8.1.1.1.7.11 element objToDataRelation/value

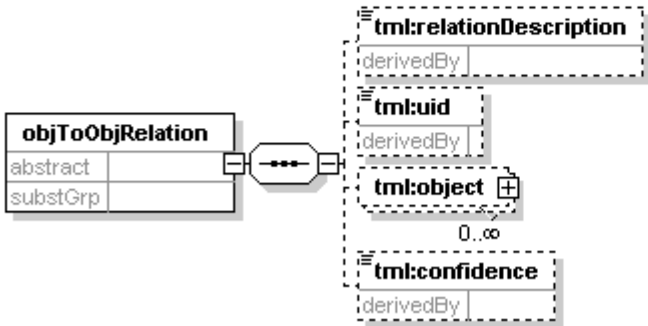
diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation A direct data value to relate to the specified object.

8.1.1.1.8 element objToObjRelation

diagram



namespace	http://www.opengis.net/tml						
properties	content	complex					
children	**tml:relationDescription** **tml:uid** **tml:object** **tml:confidence**						
used by	elements	**tml/extSysRelations** **SystemType/relations**					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group		optional			documentation	See uid_uidRef Attribute Group
	SecurityOptionsAttributeGroup		optional			documentation	See ismSecurityOptionsAttribute Group
annotation	documentation	This relation describes and subjects to subjects relations. In particular it describes relations between objects. attaching a transducer to an object (object is a subject or a transducer) (i.e. dangle, where the only thing the transducer interfaces to is that subject. The transducer to transducers relation does not include phenomenon to phenomenon connections, see dataToData					

8.1.1.1.8.1 element objToObjRelation/relationDescription

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	description of the relation					

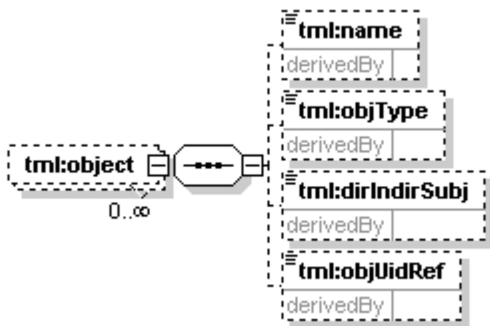
8.1.1.1.8.2 element objToObjRelation/uid

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	uid of the relationship					

8.1.1.1.8.3 element objToObjRelation/object

diagram



namespace	http://www.opengis.net/tml						
properties	isRef	0					
	minOcc	0					
	maxOcc	unbounded					
	content	complex					
children	**tml:name tml:objType tml:dirIndirSubj tml:objUidRef**						
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group
annotation	documentation	many objects can be related to a many objects. probabilities can be assigned to each relation					

8.1.1.1.8.4 element objToObjRelation/object/name

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	simple					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	name of the object					

8.1.1.1.8.5 element objToObjRelation/object/objType

diagram


namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex default subject

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: identify object as a transducer or a subject. Allowed Values: subject, transducer. Default: subject

8.1.1.1.8.6 element objToObjRelation/object/dirIndirSubj

diagram


namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex default direct

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: if objType is subject then identify if direct or indirect subject. Allowed values: direct, indirect. Default is direct.

8.1.1.1.8.7 element objToObjRelation/object/objUidRef

diagram


namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: UID of the subject (or probable subject). local id of the subject if multiple ids are used to associate with each cell of the logical structure. Sequence of values is the same as the sequence in the data (logical data structure)

8.1.1.1.8.8 element objToObjRelation/confidence

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	1

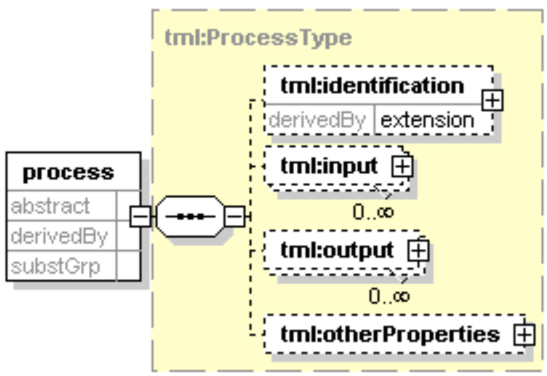
attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentation: confidence of relationship (0-1). confidence values match same sequence as logical data structure (if multiple values in data structure)

8.1.1.1.9 element process

diagram



namespace	http://www.opengis.net/tml	
type	**tml:ProcessType**	
properties	content	complex
children	**tml:identification tml:input tml:output tml:otherProperties**	
used by	elements	**tml:SystemType/processes**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group		optional			documentation	See uid_uidRef Attribute Group
SecurityOptionsAttributeGroup		optional			documentation	See ismSecurityOptionsAttribute Group

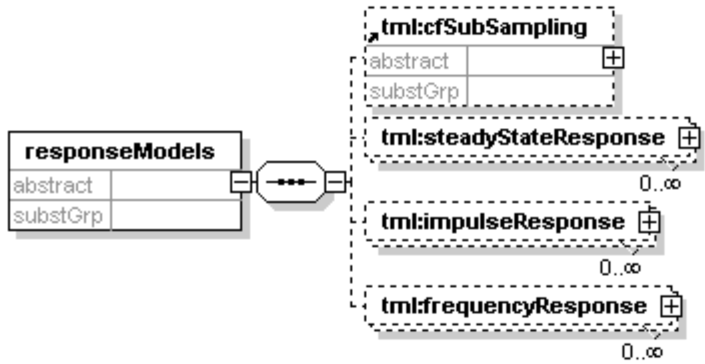
annotation — documentation: A process can be a stand alone object or part of a system. Describes derivation of output dataUnits relative to input dataUnits or constants. A process is a function that takes one or more digital inputs, and based on methodologies, produces a digital output. A process can receive data inputs and/or output data to transducers or other processes.

A TML process system is a process which inputs and/or outputs TML data. A process system does not contain transducers. The input and output to a process system is data. Either the input or the output or both of the processes must be TML data. A process algorithm is not a TML system process until either the input and/or output is in TML. Process systems can process TML data prior to being transmitted by a transmitter or after being received by a receiver.

An empty process tag in a data stream indicates that this process is no longer a part of the system

8.1.1.1.10 element responseModels

diagram



namespace	http://www.opengis.net/tml						
properties	content	complex					
children	tml:cfSubSampling tml:steadyStateResponse tml:impulseResponse tml:frequencyResponse						
used by	element	ProcessType/output/processMethods					
	complexType	TransducerType					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

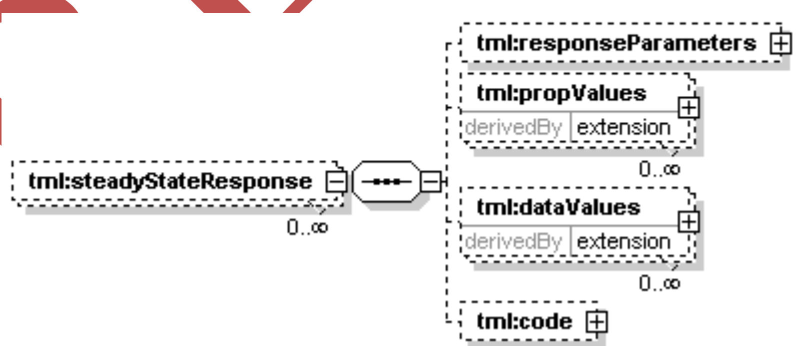
Issue Name: tml rwg 1.4 and 1.8 (M. Botts, 3 Oct 2006)

Issue Description: Derive TML behaviour models using SWE Common Parameters (particularly swe:Curve)

Resolution: Identify as an issue and document as future harmonization work with SensorML (S. Havens, 3 Oct 2006)

8.1.1.1.10.1 element responseModels/steadyStateResponse

diagram



namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded

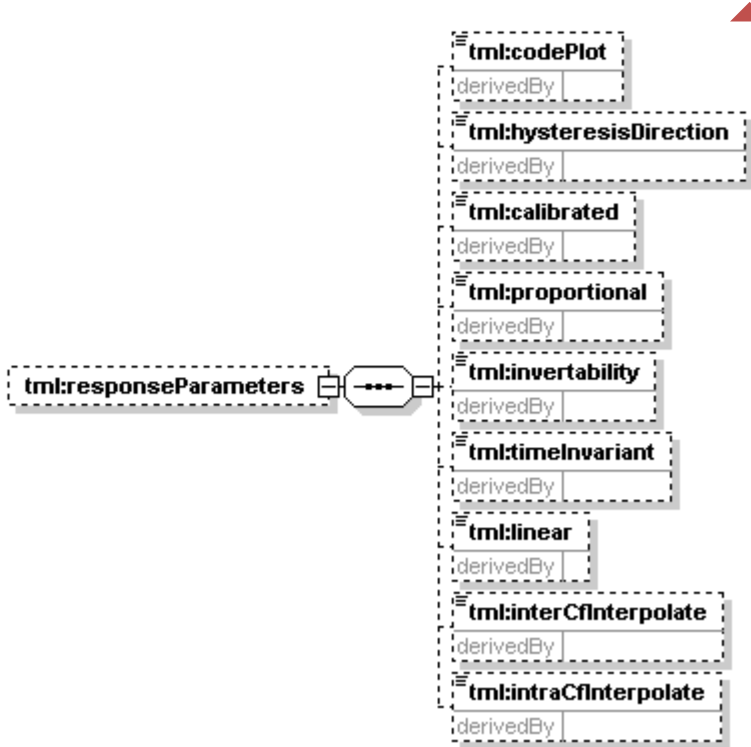
content complex
children **tml:responseParameters** **tml:propValues** **tml:dataValues** **tml:code**
attributes

Name	Type	Use	Default	Fixed	Annotation
name	**xs:string**	optional			
uid	**xs:string**	optional			documentation: uid of steady state response

annotation documentation: input to output mapping. one or more mappings for each dataUnit. Can have property-property, property-data, or data-property mappings. property-property-property and property-property-data mappings are also allowed as long as independent property values can be found somewhere. Separate mappings can be used for different hysteresis directions or for non-continuous or broken functions.

8.1.1.1.10.2 element responseModels/steadyStateResponse/responseParameters

diagram



namespace http://www.opengis.net/tml
properties
isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:codePlot** **tml:hysteresisDirection** **tml:calibrated** **tml:proportional** **tml:invertability** **tml:timeInvariant** **tml:linear** **tml:interCfInterpolate** **tml:intraCfInterpolate**

8.1.1.1.10.3 element responseModels/steadyStateResponse/responseParameters/codePlot

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties isRef 0
minOcc 0
maxOcc 1
content complex
default plot

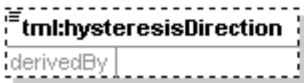
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values code, plot. Default: plot

8.1.1.1.10.4 element responseModels/steadyStateResponse/responseParameters/hysteresisDirection

diagram



namespace http://www.opengis.net/tml
type **tml:BindType**
properties isRef 0
minOcc 0
maxOcc 1
content complex
default both

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation allowed values: increasing, decreasing, both. default both

8.1.1.1.10.5 element responseModels/steadyStateResponse/responseParameters/calibrated

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties isRef 0
minOcc 0
maxOcc 1
content complex
default true

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Is response calibrated, or is response a relative reading? true of false. Default: true

8.1.1.1.10.6 element responseModels/steadyStateResponse/responseParameters/proportional

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties isRef 0
minOcc 0

	maxOcc	1
	content	complex
	default	true

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation	documentation	For uncalibrated responses is the output proportional to the input? true of false. Mult factors can also reflect prop or inversely prop for calibrated responses. Default: true.

8.1.1.1.10.7 element responseModels/steadyStateResponse/responseParameters/invertability

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	true

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	.See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation	documentation	a process input can be determined from its output. Allowed Values: true, false. default true

8.1.1.1.10.8 element responseModels/steadyStateResponse/responseParameters/timeInvariant

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	true

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation	documentation	a time shift in the input only results in a time shift in the output. Allowed Values: true, false. default true

8.1.1.1.10.9 element responseModels/steadyStateResponse/responseParameters/linear

diagram

namespace	http://www.opengis.net/tml
type	**tml:BindType**

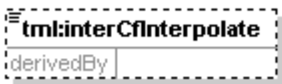
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	true

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation	documentation	allowed values: true or false. Do not need explicit phenomenon or property plot values if linear is true. Property and data mult and offset can be used if there are no limits. default true

8.1.1.1.10.10 element responseModels/steadyStateResponse/responseParameters/interCfInterpolate

diagram



namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	continuous

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation	documentation	Allowed values: continuous, discrete, lastValue, returnToZero. how to interpolate between corresponding data values between adjacent CF's. default is continuous

8.1.1.1.10.11 element responseModels/steadyStateResponse/responseParameters/intraCfInterpolate

diagram

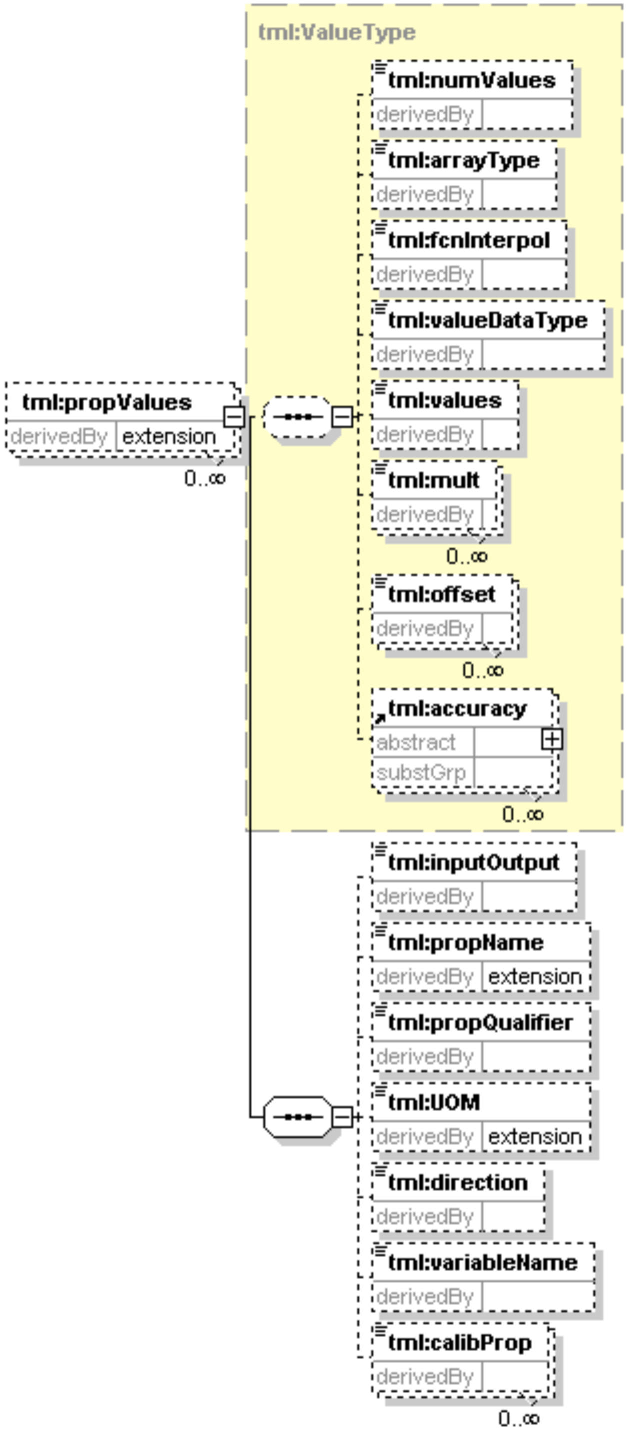
namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	continuous

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation	documentation	Allowed values: continuous, discrete, lastValue, returnToZero. how to interpolate between data values within a CF. default continuous

8.1.1.1.10.12 element responseModels/steadyStateResponse/propValues

diagram



namespace	http://www.opengis.net/tml	
type	extension of **tml:ValueType**	
properties	isRef	0
	minOcc	0

	maxOcc	unbounded
	content	complex

children: tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy tml:inputOutput tml:propName tml:UOM tml:direction tml:variableName tml:calibProp

attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation: documentation: The physical property (phenomenon) is the real world property to which the dataUnit value corresponds. This element contains the values for the physical property (phenomenon) axis of the input output transfer function

8.1.1.1.10.13 element responseModels/steadyStateResponse/propValues/inputOutput

diagram

namespace: http://www.opengis.net/tml
type: tml:BindType
properties:

isRef	0
minOcc	0
maxOcc	1
content	complex
default	input

attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	xs:string	optional			documentation: See BindType
bindUidRef	xs:string	optional			documentation: See BindType

annotation: documentation: Is the physical property (phenomenon) the input or output for this dataUnit. Allowed values: input, output. Default: input

8.1.1.1.10.14 element responseModels/steadyStateResponse/propValues/propName

diagram

namespace: http://www.opengis.net/tml
type: extension of tml:BindType
properties:

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	xs:string	optional			documentation: See BindType
bindUidRef	xs:string	optional			documentation: See BindType
uid_uidRef Attribute Group					documentation: See uid_uidRef attribute group

annotation: documentation: The propName element describes the type and identification of the phenomenon. These property (phenomenon) name values are chosen from a centralized library of phenomenology. i.e. Property (Phenomenon) Dictionary

8.1.1.1.10.15 element responseModels/steadyStateResponse/propValues/propQualifier

diagram

namespace http://www.opengis.net/tml

type extension of **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType
uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

annotation documentation Qualifier for the property. From Qualifier Dictionary. e.g. avgValue, rmsValue, rssValue, instValue, accumulatedValue, rateOfChange, range, min, max...

Issue Name: tml rwg 1.14 (S. Cox, 3 Oct 2006)

Issue Description: Discussion of standardization of phenomena/property dictionary needs references to relevant OGC specs that deal with this topic - O&M &/or SWE architecture

Resolution: The intent is to adopt Phenomena Dictionary URNs when available (S. Havens, 3 Oct 2006)

8.1.1.1.10.16 element responseModels/steadyStateResponse/propValues/UOM

diagram

namespace http://www.opengis.net/tml

type extension of **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	see BindType
bindUidRef	**xs:string**	optional			documentation	see BindType
uid_uidRef attribute group	**xs:string**	optional			documentation	see uid_uidRef attribute group

annotation documentation This is from the Units Of Measure Dictionary (SI Units) which describes the unit of measure of the particular property or phenomena .

8.1.1.1.10.17 element responseModels/steadyStateResponse/propValues/direction

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex
default	none

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation if the physical property (phenomenon) had a direction associated with it such as torque or force. direction relative to the transducer reference system. Allowed Values: horizontal, vertical, +xTranslation, -xTranslation, +yTranslation, -yTranslation, +zTranslation, -zTranslation, +alpha, -alpha, +beta, -beta, +rhoTranslation, -rhoTranslation, +latTranslation, -latTranslation, +longTranslation -longTranslation, +altTranslation, -altTranslation, +omegaRotation, -omegaRotation, +phiRotation, -phiRotation, +kappaRotation, -kappaRotation, none Default: none

8.1.1.1.10.18 element responseModels/steadyStateResponse/propValues/variableName

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

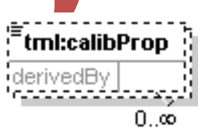
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation .See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Name of mathematical term used in the transformation equations.

8.1.1.1.10.19 element responseModels/steadyStateResponse/propValues/calibProp

diagram



namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	unbounded
content	complex
default	none

attributes

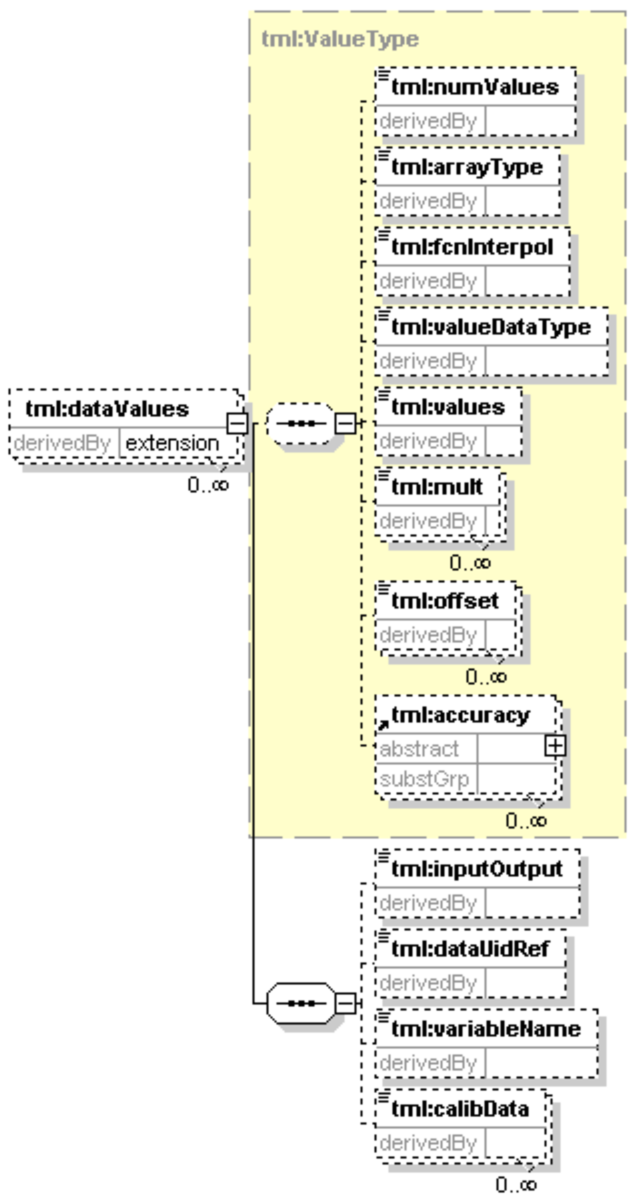
Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType

	bindUidRef	**xs:string**	optional	documentation	See BindType

annotation documentation This is the calibrated value of a calibrated source providing a calibrated stimulus to a sensor across its TCF array, or it is the response from a data value stimulus for a calibrated transmitter or actuator across its TCF array. Typically it may just be a single value (e.g. black body source). There may be multiple calibration points within the dynamic range of the transducer. The calibration source can be from an external calibrated sensor measuring the stimulus. If a calibrated source is available this elements identifies the calibration level or points (bindUID) to the calibrated sensor measuring the source. This is used for post correcting relative readings Default: none

8.1.1.1.10.20 element responseModels/steadyStateResponse/dataValues

diagram



namespace http://www.opengis.net/tml
type extension of **tml:ValueType**
properties isRef 0
minOcc 0

maxOcc unbounded
content complex

children **tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy tml:inputOutput tml:dataUidRef tml:variableName tml:calibData**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation documentation: This element contains the values for the data axis of the input output transfer function. For linear functions the slope is determined by the two multipliers. If the mult on data and prop are opposite then slope is negative otherwise it is positive. Axis offsets are determined by offset values

8.1.1.1.10.21 element responseModels/steadyStateResponse/dataValues/inputOutput

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
default output

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation: See BindType
bindUidRef	**xs:string**	optional			documentation: See BindType

annotation documentation: Is the data an input or an output for this dataUnit. Allowed values: input, output. Default: output

8.1.1.1.10.22 element responseModels/steadyStateResponse/dataValues/dataUidRef

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation: See BindType
bindUidRef	**xs:string**	optional			documentation: See BindType

annotation documentation: uid of the data form the logical data structure (dataUnit) to which this response model corresponds

8.1.1.1.10.23 element responseModels/steadyStateResponse/dataValues/variableName

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties isRef 0
minOcc 0
maxOcc 1
content complex

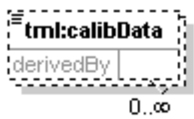
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Name of mathematical term used in the transformation equations. This is the same as the variableName used in the referenced dataUnit

8.1.1.1.10.24 element responseModels/steadyStateResponse/dataValues/calibData

diagram



namespace http://www.opengis.net/tml
type **tml:BindType**
properties isRef 0
minOcc 0
maxOcc unbounded
content complex
default none

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation This is the data resulting from calibrated source. or bindUID points to sensor measurement measuring calib source. Default: none

8.1.1.1.10.25 element responseModels/steadyStateResponse/code

diagram

namespace http://www.opengis.net/tml
properties isRef 0
minOcc 0
maxOcc 1
content complex
children **properties listing**

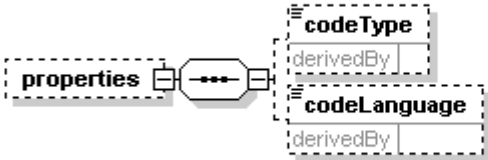
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation See uid_uidRef Attribute Group

annotation documentation computer code of the transfer process from input to output

8.1.1.1.10.26 element responseModels/steadyStateResponse/code/properties

diagram



namespace http://www.opengis.net/tml

properties
isRef 0
minOcc 0
maxOcc 1
content complex

children **codeType** **codeLanguage**

8.1.1.1.10.27 element responseModels/steadyStateResponse/code/properties/codeType

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default source

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed Values: source, exe default: source

8.1.1.1.10.28 element responseModels/steadyStateResponse/code/properties/codeLanguage

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default C

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed Values: C, C++, Java, Fortran, C Sharp, Basic, Visual Basic. Default: C

8.1.1.1.10.29 element responseModels/steadyStateResponse/code/listing

diagram

namespace http://www.opengis.net/tml
type extension of **tml:BindType**
properties isRef 0
minOcc 0
maxOcc 1
content complex

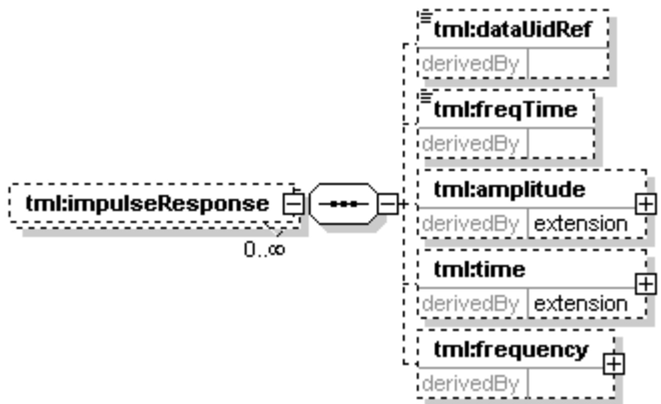
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation .See BindType

annotation documentation Listing of code. Base64 encoded executable or source code with unallowed XML characters escaped out

8.1.1.1.10.30 element responseModels/impulseResponse

diagram



namespace http://www.opengis.net/tml
properties isRef 0
minOcc 0
maxOcc unbounded
content complex
children **tml:dataUidRef tml:freqTime tml:amplitude tml:time tml:frequency**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation The impluseResponse element describes the output of a linear time invariant transform. It is expressed as a function of time or complex frequency which is the result of an impulse of infinite magnitude and infinitesimal duration. May have a separate response for each dataUnit and for each type (freq and time). Or dataUnits within a data Set may share the same response.

8.1.1.1.10.31 element responseModels/impulseResponse/dataUidRef

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties isRef 0
minOcc 0
maxOcc 1
content complex
attributes Name Type Use Default Fixed Annotation

	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentation: same as uidRef in attributes

8.1.1.1.10.32 element responseModels/impulseResponse/freqTime

diagram

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties:

isRef	0
minOcc	0
maxOcc	1
content	complex
default	time

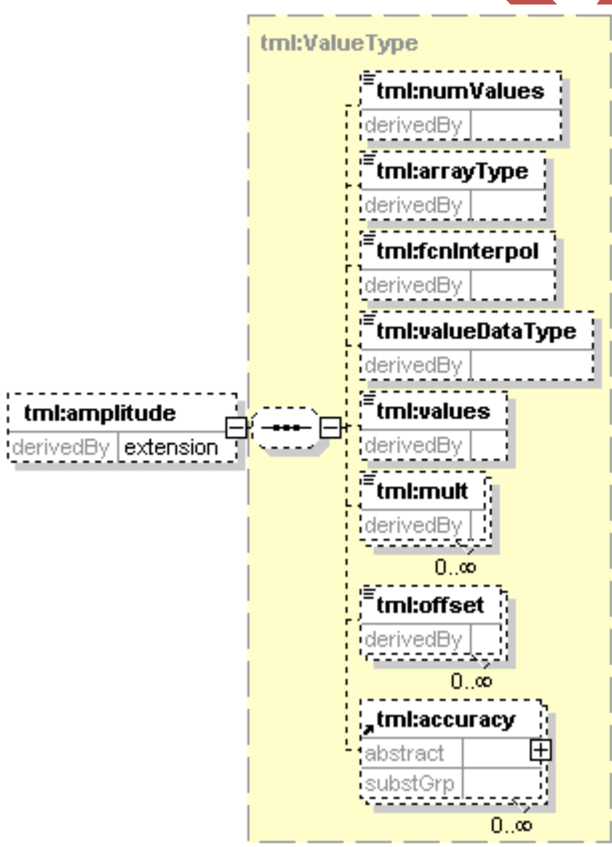
attributes:

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentation: Allowed values: freq, time. default is time. indicates if frequency or time domain descriptions.

8.1.1.1.10.33 element responseModels/impulseResponse/amplitude

diagram



namespace: http://www.opengis.net/tml

type: extension of **tml:ValueType**

properties:

isRef	0

minOcc 0
maxOcc 1
content complex

children [tml:numValues] [tml:arrayType] [tml:fcnInterpol] [tml:valueDataType] [tml:values] [tml:mult] [tml:offset] [tml:accuracy]

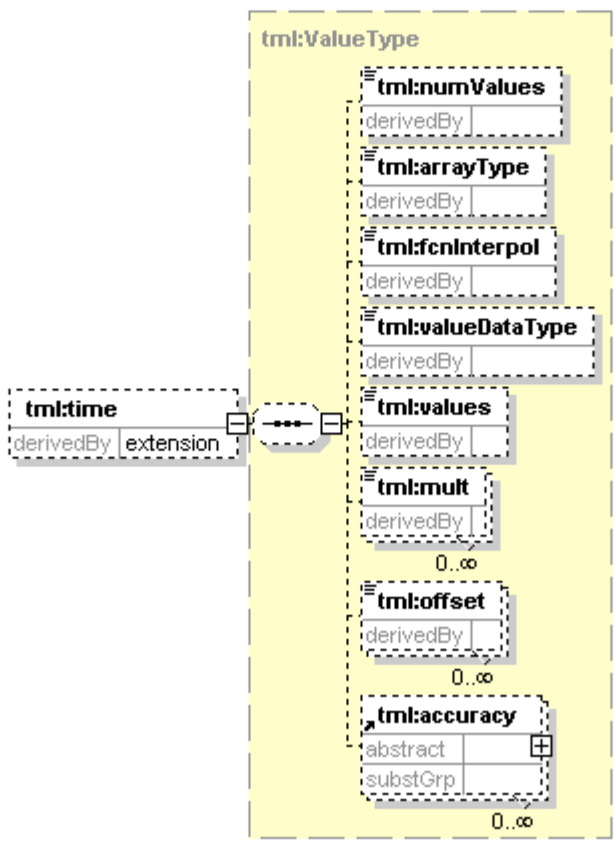
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation documentation amplitude dependent axis.

8.1.1.1.10.34 element responseModels/impulseResponse/time

diagram



namespace http://www.opengis.net/tml
type extension of tml:ValueType
properties
isRef 0
minOcc 0
maxOcc 1
content complex

children tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy

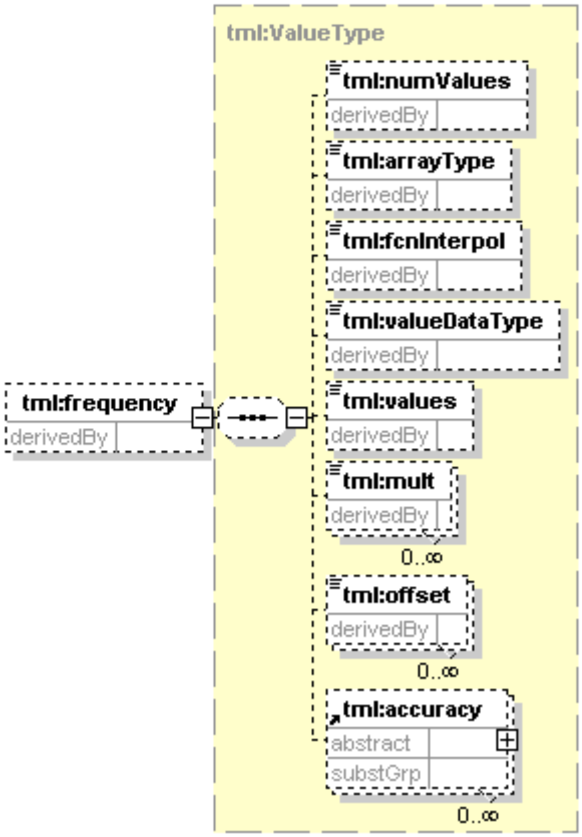
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation documentation time domain independent axis.

8.1.1.1.10.35 element responseModels/impulseResponse/frequency

diagram



namespace: http://www.opengis.net/tml
type: **tml:ValueType**
properties:
- isRef 0
- minOcc 0
- maxOcc 1
- content complex

children: **tml:numValues** **tml:arrayType** **tml:fcnInterpol** **tml:valueDataType** **tml:values** **tml:mult** **tml:offset** **tml:accuracy**

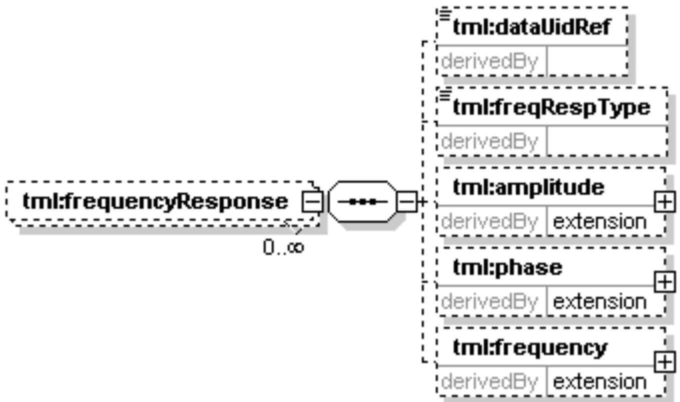
attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation: documentation: frequency domain independent axis.

8.1.1.1.10.36 element responseModels/frequencyResponse

diagram



namespace http://www.opengis.net/tml

properties isRef 0
minOcc 0
maxOcc unbounded
content complex

children **tml:dataUidRef** **tml:freqRespType** **tml:amplitude** **tml:phase** **tml:frequency**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

annotation documentation This element characterized the response as a function of frequency. (i.e. carrier) It also describes any secondary modulation of the carrier waveform. For transmitters it describes the instantaneous power spectral density. The frequency response will be characterized as either carrier, modulation or power spectral density type. A separate response function can be described for each type and for each dataUnit and each type of plot amp vs. freq and phase vs. freq (can combine plots onto one as well)

8.1.1.1.10.37 element responseModels/frequencyResponse/dataUidRef

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation same as uidRef in attributes

8.1.1.1.10.38 element responseModels/frequencyResponse/freqRespType

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

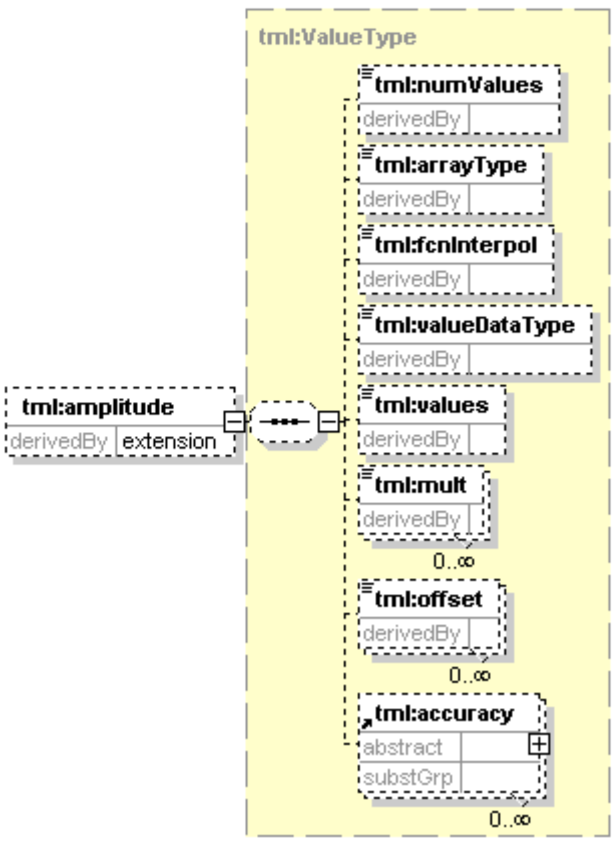
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	carrier

attributes	Name	Type	Use	Default	Fixed	Annotation
	bindUid	**xs:string**	optional			documentation See BindType
	bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values: carried, modulation, PSD (pwrSpectralDensity). default carrier

8.1.1.1.10.39 element responseModels/frequencyResponse/amplitude

diagram



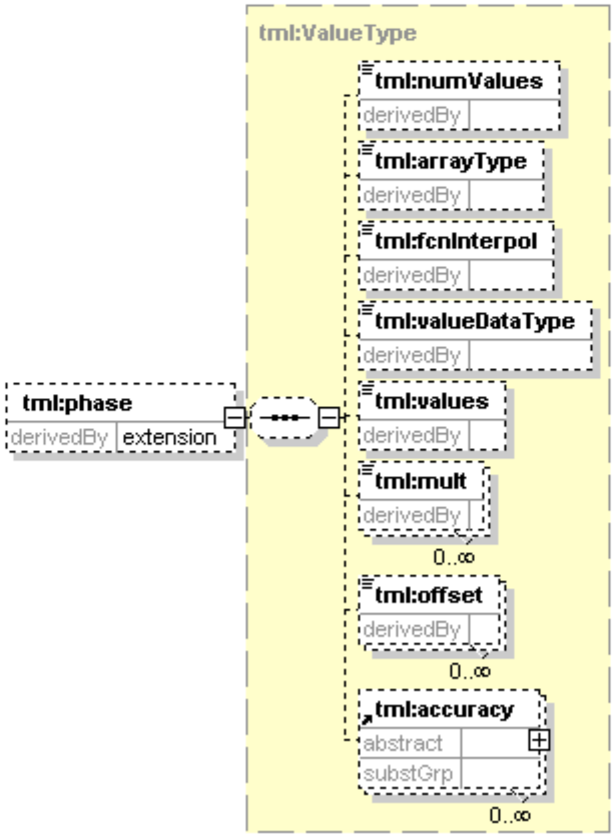
namespace	http://www.opengis.net/tml	
type	extension of **tml:ValueType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy**	

attributes	Name	Type	Use	Default	Fixed	Annotation
	uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation Set of point coordinates describing amplitude dependent axis

8.1.1.1.10.40 element responseModels/frequencyResponse/phase

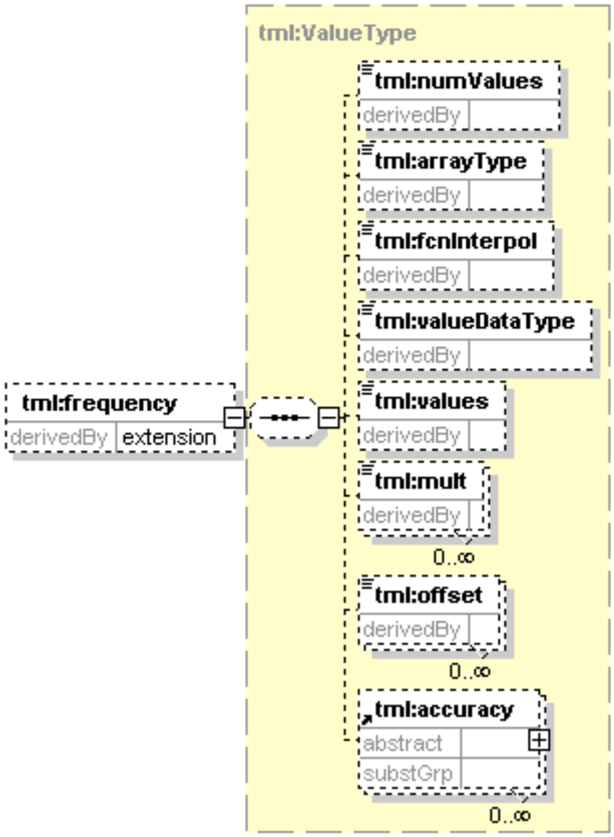
diagram



namespace	http://www.opengis.net/tml						
type	extension of **tml:ValueType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
children	**tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy**						
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group
annotation	documentation	Set of point coordinates describing phase dependent axis					

8.1.1.1.10.41 element responseModels/frequencyResponse/frequency

diagram



namespace http://www.opengis.net/tml
type extension of **tml:ValueType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:numValues** **tml:arrayType** **tml:fcnInterpol** **tml:valueDataType** **tml:values** **tml:mult** **tml:offset** **tml:accuracy**

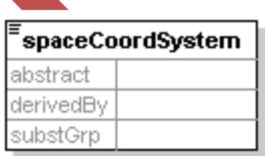
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation Set of point coordinates describing frequency independent axis

8.1.1.1.11 element spaceCoordSystem

diagram



namespace http://www.opengis.net/tml
type **tml:BindType**
properties
content complex
default spherical
used by element **TransducerType/spatialModel/ambiguitySpace/shape**

complexType **SpatialCoordType**

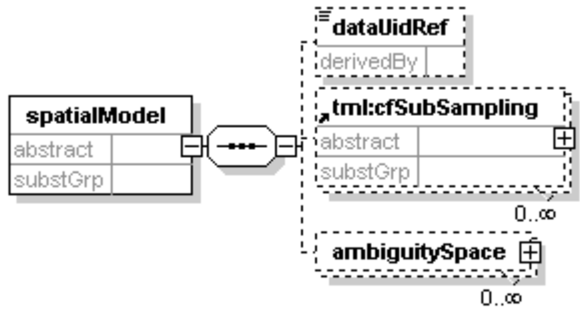
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values: spherical, rectangular, cylindrical, wgs84elliptical. default is spherical.

8.1.1.1.12 element spatialModel

diagram



namespace http://www.opengis.net/tml

properties content complex

children **dataUidRef** **tml:cfSubSampling** **ambiguitySpace**

used by element **ProcessType/output**

complexType **TransducerType**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation See uid_uidRef Attribute Group

8.1.1.1.12.1 element spatialModel/dataUidRef

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef 0
minOcc 0
maxOcc 1
content complex

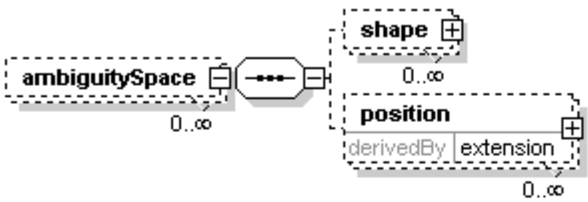
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation corresponding UID of dataUnit, dataSet, or data Array. If data array then all subordinate data structures share same model (row, col, or plane), if dataSet then all data units share same model (cf), if dataUnit then only that units model is described (cf).

8.1.1.1.12.2 element spatialModel/ambiguitySpace

diagram



namespace	http://www.opengis.net/tml
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	[shape](#) [position](#)

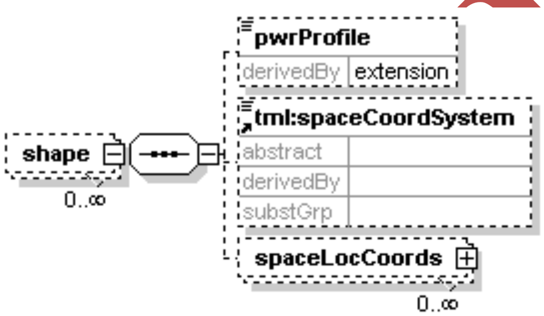
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation: See uid_uidRef Attribute Group

annotation documentation: Multiple AS are combined as spatial intersections. e.g. one for columns and one for rows. Typically every cell within a multiple cell CF will share the same shape but have unique positions.

8.1.1.1.12.3 element spatialModel/ambiguitySpace/shape

diagram



namespace	http://www.opengis.net/tml
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	[pwrProfile](#) [tml:spaceCoordSystem](#) [spaceLocCoords](#)

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation: See uid_uidRef Attribute Group

annotation documentation: This is the shape of the AS for the power profile indicated. May also have multiple shapes to define multiple lobes of energy fields. Multiple shapes within an AS are combined as a spatial unions. The position elements defines the position of each shape.

8.1.1.1.12.4 element spatialModel/ambiguitySpace/shape/pwrProfile

diagram

namespace	http://www.opengis.net/tml
type	extension of [tml:BindType](#)

properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	default	-3

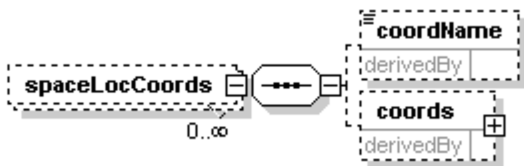
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation The equi-power surface power level compared to the point of transmission or reception. -3db beam pattern, pwrProfile="-3".

8.1.1.1.12.5 element spatialModel/ambiguitySpace/shape/spaceLocCoords

diagram



namespace http://www.opengis.net/tml

properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex

children **coordName** **coords**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation See uid_uidRef Attribute Group

annotation documentation one set of coordinates for each spatial axes. Each shape is defined relative to an arbitrary data spatial reference system.

8.1.1.1.12.6 element spatialModel/ambiguitySpace/shape/spaceLocCoords/coordName

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

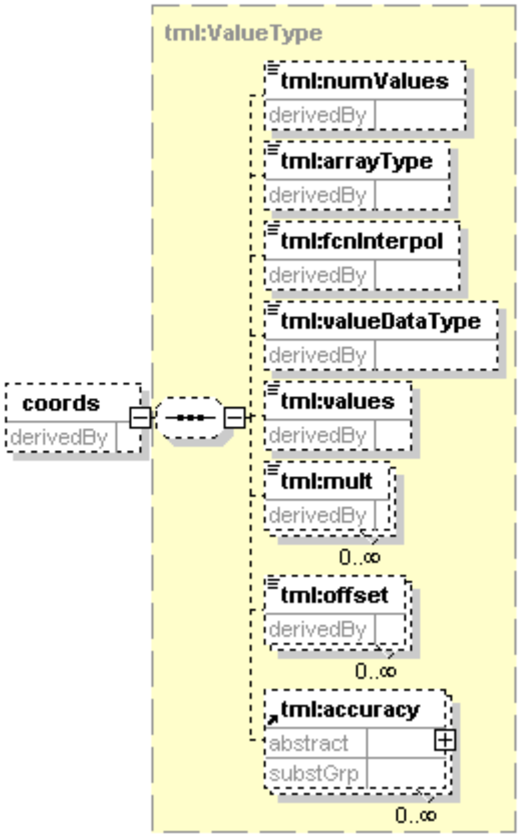
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed values: x, y, z, alpha, beta, rho.

8.1.1.1.12.7 element spatialModel/ambiguitySpace/shape/spaceLocCoords/coords

diagram



namespace http://www.opengis.net/tml

type **tml:ValueType**

properties isRef 0
minOcc 0
maxOcc 1
content complex

children **tml:numValues** **tml:arrayType** **tml:fcnInterpol** **tml:valueDataType** **tml:values** **tml:mult** **tml:offset** **tml:accuracy**

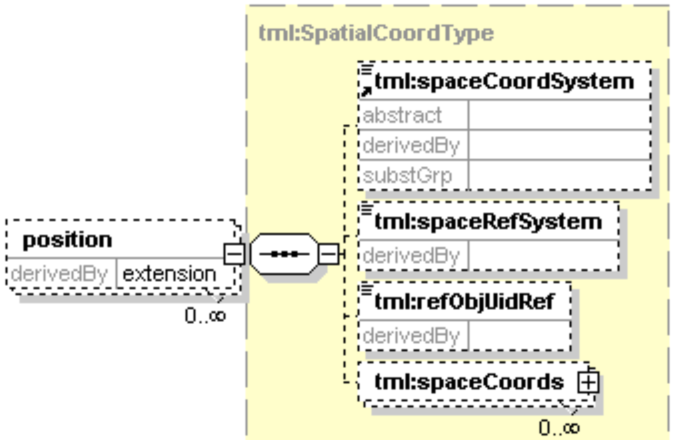
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation: See uid_uidRef Attribute Group

annotation documentation values contains a string of real numbers. The mult and offset are single values, unless the shape varies over the Characteristic Frame then the mult and offset may contain a Characteristic Frame array of values. simple IFOV alpha=0, beta=0. (ray where rho is infinite)

8.1.1.1.12.8 element spatialModel/ambiguitySpace/position

diagram



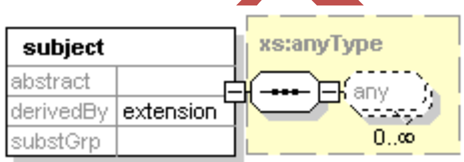
namespace http://www.opengis.net/tml
type extension of **tml:SpatialCoordType**
properties
isRef 0
minOcc 0
maxOcc unbounded
content complex
children **tml:spaceCoordSystem tml:spaceRefSystem tml:refObjUidRef tml:spaceCoords**
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation See uid_uidRef Attribute Group

annotation documentation location and attitude of ambiguity shape

8.1.1.1.13 element subject

diagram



namespace http://www.opengis.net/tml
type extension of **xs:anyType**
properties content complex
used by elements **tml SystemType/subjects**
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation This is the subject (object, thing) that relates to the phenomenon (property) that is affected or detected by the transducer. For example, if one is measuring the temperature on an exhaust manifold, the transducer is the thermocouple, and the object is the manifold. This adds extra context to the data. The relation between a subject and transducer data or subject and subject is described in the relationship element. An empty subject tag in a data stream indicates that this object is no longer a part of the system

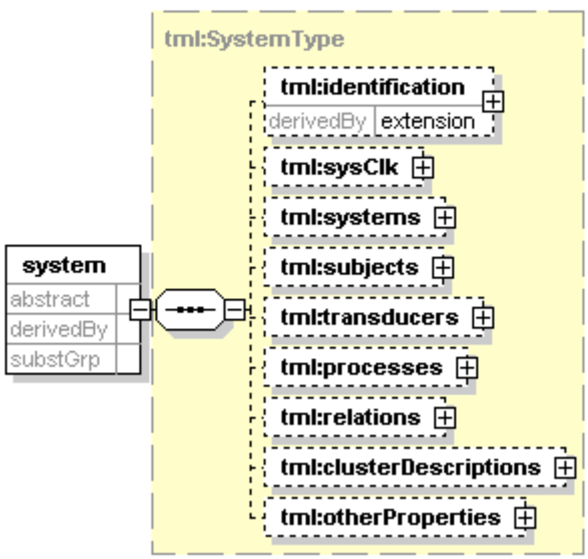
Issue Name: tml rwg 1.24 (S. Cox, 3 Oct 2006)

Issue Description: The XML encoding is not derived from GML. This is noted as a future goal. However, this could be made easier now at little cost by following the GML encoding patterns now. In particular through use of (i) element-content in preference to XML attributes for data, (ii) normalization of common objects with use of xlinks for cross-referencing, supported by URI's rather than text for identifiers, and (iii) interleaving elements representing Objects and properties. The latter would also be made easier by use of UML following the ISO 19103 profile for system modeling. Note that this maps quite directly into XML Spy diagrams. The coding standards in the XML Schemas, particularly concerning global type definitions and element declarations are also inconsistent with conventional practice in OGC data standards

Resolution: Identify as an issue and document as future harmonization work with GML and ISO Specifications (S. Havens, 3 Oct 2006)

8.1.1.1.14 element system

diagram



namespace: http://www.opengis.net/tml

type: tml:SystemType

properties: content complex

children: tml:identification tml:sysClk tml:systems tml:subjects tml:transducers tml:processes tml:relations tml:clusterDescriptions tml:otherProperties

used by: elements tml SystemType/systems

attributes:

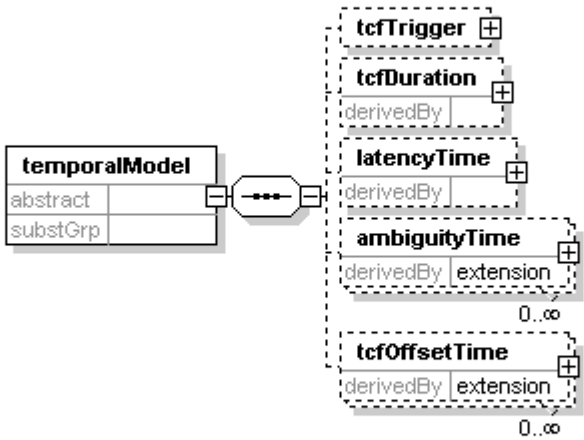
Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group		optional			documentation	See uid_uidRef Attribute Group
SecurityOptionsAttributeGroup		optional			documentation	See ismSecurityOptionsAttribute Group

annotation: documentation: A system is composed of one or more objects, transducers, relations, processes or cluster descriptions. The system description gives logical syntax and context to the data produced by the system. It describes the transducer organization of data, the transducer response and geometry characteristics, the pre-processing between the data and the transducer, and finally it describes the external logical relationships relating to understanding how multiple

components (transducers and processes) within a system relate to one another. Transducer systems are a collection of transducers and processes working together to produce TML data or to utilize TML data to invoke desired phenomenon states. Transducers in a system may be independent or complementary (provide supporting measurements for other transducers).
EXAMPLE: A complementary set of transducers: a GPS sensor provides position and an IMU sensor provides attitude for a camera sensor.
An empty system tag (with id) in a data stream indicates that the system is no longer available in the stream, or if system was not previously part of the parent system it will be added to the parent system.

8.1.1.1.15 element temporalModel

diagram



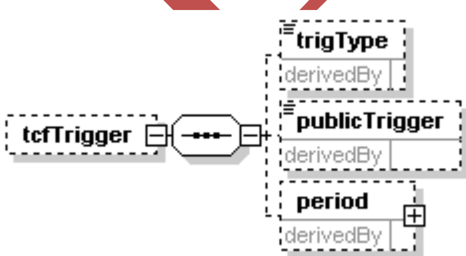
namespace	http://www.opengis.net/tml
properties	content complex
children	**tcfTrigger tcfDuration latencyTime ambiguityTime tcfOffsetTime**
used by	element **ProcessType/output** complexType **TransducerType**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

8.1.1.1.15.1 element temporalModel/tcfTrigger

diagram



namespace	http://www.opengis.net/tml
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	**trigType publicTrigger period**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef					documentation See uid_uidRef Attribute

Attribute Group					Group

8.1.1.1.15.2 element temporalModel/tcfTrigger/trigType

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex
default	privatePeriodic

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed Values: private, privateOnDataRecipt, privateOnInputTrig, pvtOnChgOutput. publicOnTrigReciept. public trigger: controllable by external commands. private trigger: uncontrollable by external commands. Virtual trig sensor puts sysClk time in data tag. If public a bindUid is made available. default trigger is privatePeriodic.

8.1.1.1.15.3 element temporalModel/tcfTrigger/publicTrigger

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

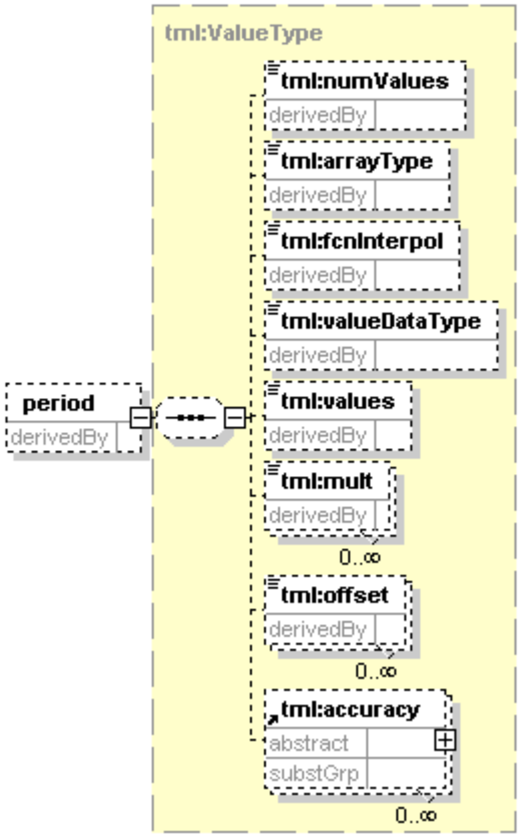
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation if trigger is public then this identifies the uidRef of trigger source (command). Whenever a data cluster is sent to this UID or to the uid of a process that is bound to this uid then this transducer or process cycle will trigger. The bindUid enables late binding of the trigger source

8.1.1.1.15.4 element temporalModel/tcfTrigger/period

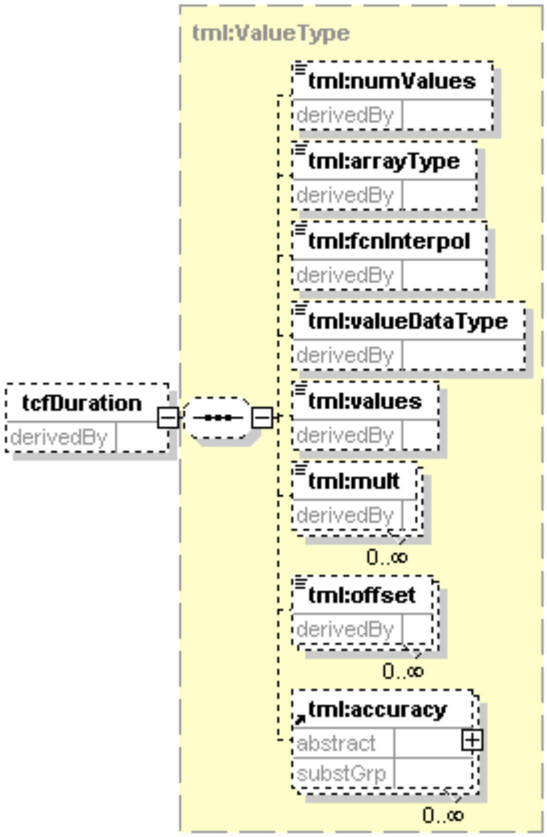
diagram



namespace	http://www.opengis.net/tml
type	tml:ValueType
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy
attributes	Name: uid_uidRef Attribute Group; Type; Use; Default; Fixed; Annotation: documentation See uid_uidRef Attribute Group
annotation	documentation: if private trigger is periodic then, trigger period in seconds

8.1.1.1.15.5 element temporalModel/tcfDuration

diagram



namespace	http://www.opengis.net/tml
type	**tml:ValueType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	**tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy**

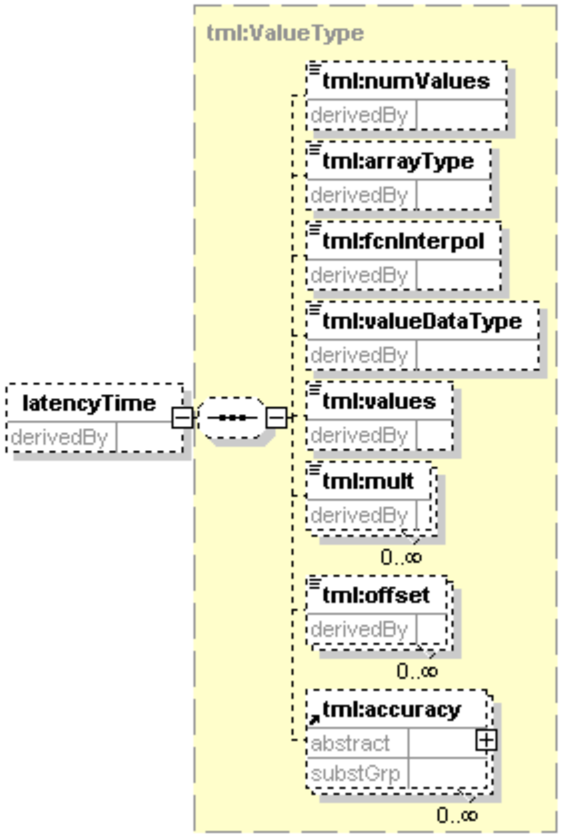
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation documentation: time duration of the TCF in seconds. Can also be determined by the TCF offset time values by subtracting the smallest offset time from the largest offset time. Duration does not vary over the TCF. There is only one value.

8.1.1.1.15.6 element temporalModel/latencyTime

diagram



namespace http://www.opengis.net/tml

type **tml:ValueType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:numValues** **tml:arrayType** **tml:fcnInterpol** **tml:valueDataType** **tml:values** **tml:mult** **tml:offset** **tml:accuracy**

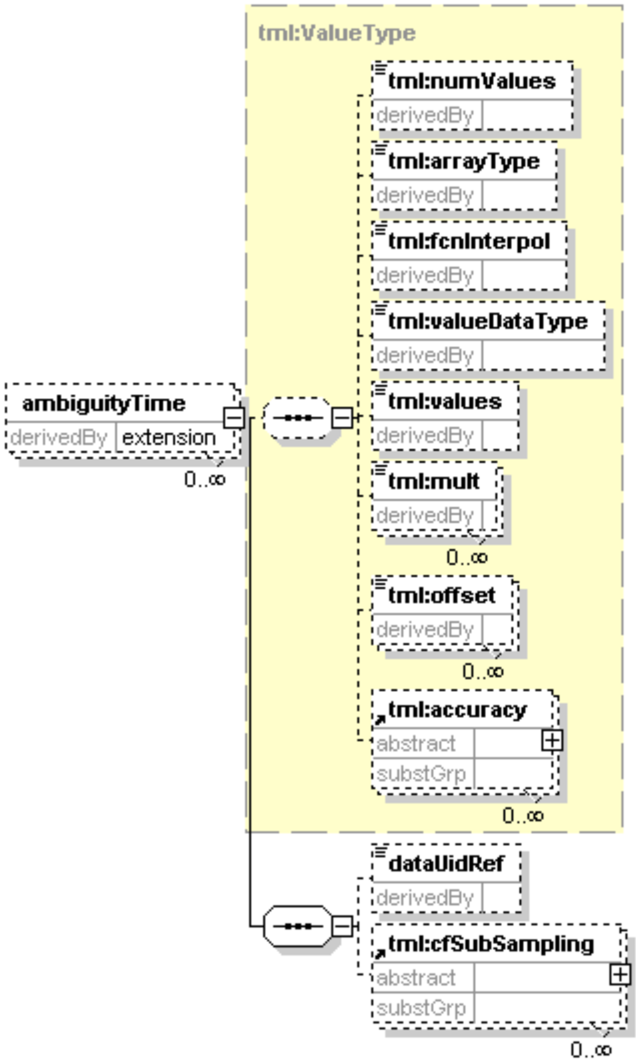
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation documentation latency time in seconds (real number). Time between the input and the output. Transducer time tags should be corrected to reflect correct input time for receivers and output time for transmitters. Latency for processes reflects the process delay. Latency time does not vary over the CF. There is only one value.

8.1.1.1.15.7 element temporalModel/ambiguityTime

diagram



namespace	http://www.opengis.net/tml
type	extension of **tml:ValueType**
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	**tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy dataUidRef tml:cfSubSampling**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation documentation: data integration time for each sample in the TCF. Each dataunit may have a different time. This element contains the number of samples in a TCF or the number indicated by the noOfSubSampledIndexPoints element in the TCFsubSamplingSequence or just one time. If just one time then the same time applies to all sample in the TCF.

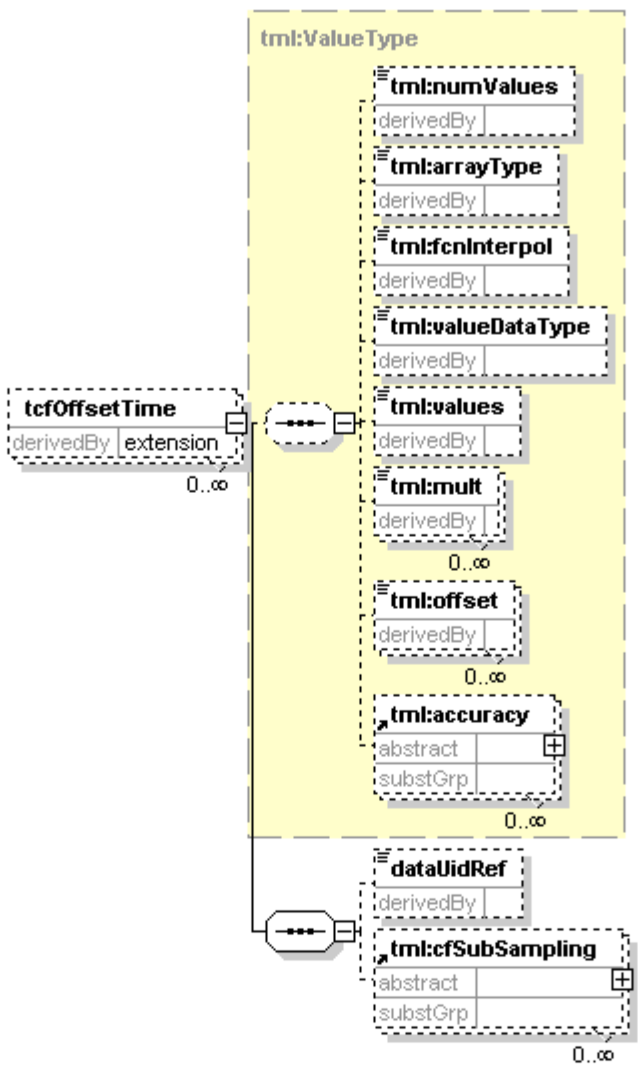
8.1.1.1.15.8 element temporalModel/ambiguityTime/dataUidRef

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	corresponding UID of dataUnit or dataSet. Duplicate of uid in identification element Default: Uid of dataSet					

8.1.1.1.15.9 element temporalModel/tcfOffsetTime

diagram



namespace http://www.opengis.net/tml

type: extension of **tml:ValueType**

properties:

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children: **tml:numValues** **tml:arrayType** **tml:fcnInterpol** **tml:valueDataType** **tml:values** **tml:mult** **tml:offset** **tml:accuracy** **dataUidRef** **tml:cfSubSampling**

attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation: documentation: tcfOffSetTime contains time offsets for each dataUnit or dataSet in the TCF relative to the clock attribute (clk or dateTime) in the data start tag. contains the number of time values indicated by the numSubSampledIndexPoints in the cfSubSampling child element. or num

8.1.1.1.15.10 element temporalModel/tcfOffsetTime/dataUidRef

diagram:

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties:

isRef	0
minOcc	0
maxOcc	1
content	complex

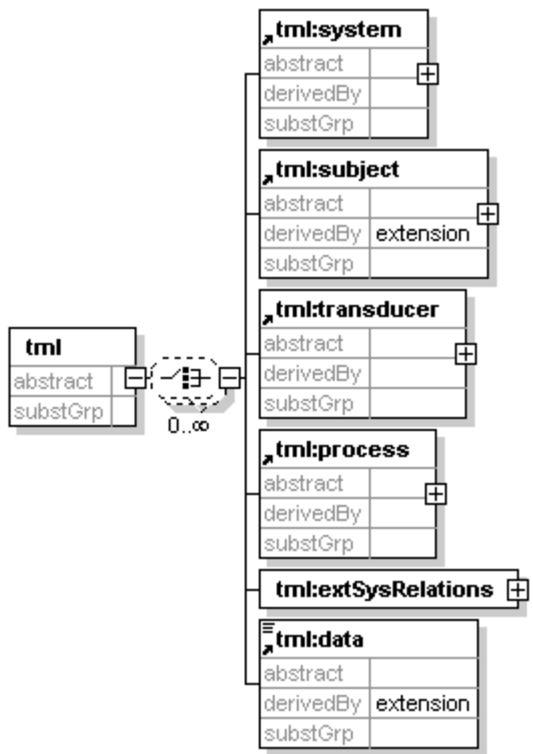
attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation: See BindType
bindUidRef	**xs:string**	optional			documentation: See BindType

annotation: documentation: corresponding UID of dataUnit or dataSet. Duplicate of uid in identification element Default: Uid of dataSet

8.1.1.1.16 element tml

diagram



namespace http://www.opengis.net/tml

properties content complex

children tml:system tml:subject tml:transducer tml:process tml:extSysRelations tml:data

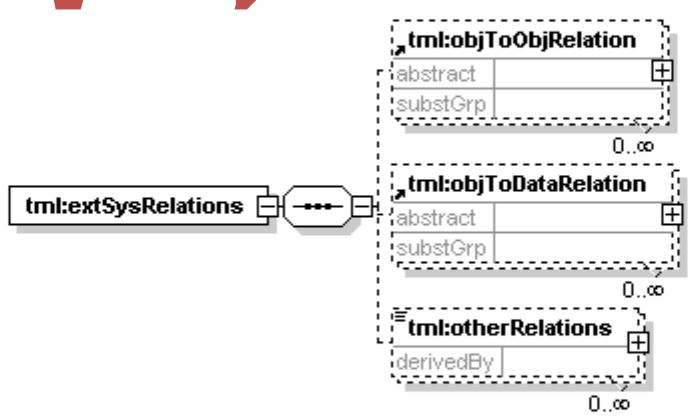
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group		optional			documentation See uid_uidRef Attribute Group
version	**xs:string**	required		1.0	documentation fixed version 1.0
ismSecurityOptionsAttributeGroup		optional			See ismSecurityOptionsAttributeGroup

annotation documentation Root Element. Also contains attributes to indicate the overall security classification of this TML stream or file. If needed individual data clusters can be labelled with a security class.

8.1.1.1.16.1 element tml/extSysRelations

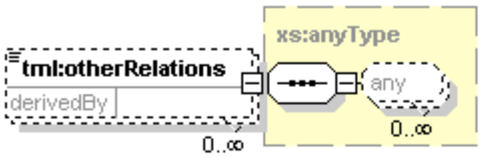
diagram



namespace http://www.opengis.net/tml
properties isRef 0
content complex
children tml:objToObjRelation tml:objToDataRelation tml:otherRelations
annotation documentation for relating external subject to external subject or transducer data to external subject. An external subject(object) is external to the system.

8.1.1.1.16.2 element tml/extSysRelations/otherRelations

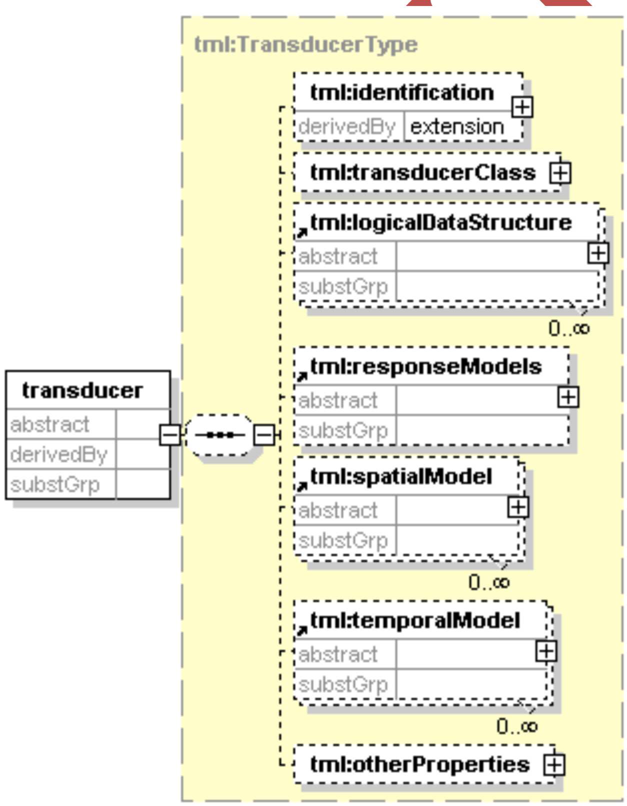
diagram



namespace http://www.opengis.net/tml
type **xs:anyType**
properties isRef 0
minOcc 0
maxOcc unbounded
content complex
mixed true
attributes Name Type Use Default Fixed Annotation

8.1.1.1.17 element transducer

diagram



namespace http://www.opengis.net/tml
type tml:TransducerType
properties content complex

children	**tml:identification** **tml:transducerClass** **tml:logicalDataStructure** **tml:responseModels** **tml:spatialModel** **tml:temporalModel** **tml:otherProperties**						
used by	elements	**tml SystemType/transducers**					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group		optional			documentation	See uid_uidRef Attribute Group
	SecurityOptionsAttributeGroup		optional			documentation	See ism:SecurityOptionsAttributeGroup
annotation	documentation	Transducers that translate phenomenon to data are typically referred to as receivers or sensors, and transducers that translate data to phenomenon are typically referred to as transmitters or actuators. Transducers are the interface between the digital realm and the physical realm or real world. Any time a computing machine is processing data, its input is received through a transducer and its output is through a transducer. A transducer can be a stand alone object or part of a system. An empty transducer tag in a data stream indicates that this transducer is no longer a part of the system					

Issue Name: tml rwg 1.3 and 1.7 (M. Botts, 3 Oct 2006)

Issue Description: Recommend that TML derive transducer from sml:ProcessModel. During OWS3, we demonstrated through an example schema how tml:transducer could be derived from SensorML. We feel that such a derivation using the SensorML framework would provide complementary TML-SensorML schema, and would allow tml:transducer to be used directly within SensorML sensor descriptions and process chains.

Resolution: Identify as an issue and document as future harmonization work with SensorML (S. Havens, 3 Oct 2006)

8.1.1.2 TML Complex Types

The following section contains descriptions of the complex types used in TransducerML, The complex types are described in the following order.

- **BindType**
- **DataArrayType**
- **IdentificationType**
- **ProcessType**
- **SpatialCoordType**
- **SystemType**
- **TransducerType**
- **ValueType**

8.1.1.2.1 complexType BindType

diagram

namespace http://www.opengis.net/tml
type extension of **xs:anySimpleType**

properties	base	xs:stringType
used by	elements	responseModels/steadyStateResponse/code/properties/codeLanguage responseModels/steadyStateResponse/code/properties/codeType spatialModel/ambiguitySpace/shape/spaceLocCoords/coordName temporalModel/tcfOffsetTime/dataUidRef temporalModel/ambiguityTime/dataUidRef spatialModel/dataUidRef responseModels/steadyStateResponse/code/listing temporalModel/tcfTrigger/publicTrigger spatialModel/ambiguitySpace/shape/pwrProfile spaceCoordSystem SystemType/relations/timeRelation/timeCoordinate/absTimeUidRef accuracy/accyValues DataArrayType/arrayOf ValueType/arrayType clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize/beginTextDelimiter clusterDesc/dataUnitEncoding/dataUnitFieldSize/beginTextDelimiter dataUnit/bytesInBlob responseModels/steadyStateResponse/dataValues/calibData responseModels/steadyStateResponse/propValues/calibProp responseModels/steadyStateResponse/responseParameters/calibrated cfSubSampling/cfStructComp clusterDesc/clusterProperties/clusterType responseModels/steadyStateResponse/responseParameters/codePlot clusterDesc/clusterProperties/complexity objToDataRelation/object/confidence objToObjRelation/confidence SpatialCoordType/spaceCoords/coordName SystemType/sysClk/countNumBase clusterDesc/binHeaderEncode/headerAttrib/dataType clusterDesc/dataUnitEncoding/dataType dataUnit/dataType SystemType/relations/dataToDataRelation/dataSink/dataUidRef SystemType/relations/dataToDataRelation/dataSource/dataUidRef responseModels/steadyStateResponse/dataValues/dataUidRef objToDataRelation/dataUidRef responseModels/frequencyResponse/dataUidRef responseModels/impulseResponse/dataUidRef clusterDesc/dataUnitEncoding/dataUnitUidRef ProcessType/input/dataValue IdentificationType/characterization/characterizedBy/date IdentificationType/characterization/validatedBy/date ProcessType/identification/ownedBy/date TransducerType/identification/ownedBy/date SystemType/identification/owner/date IdentificationType/calibration/calibratedBy/date SystemType/identification/operator/date IdentificationType/calibration/validatedBy/date ProcessType/input/inputIdent/description ProcessType/output/outputIdent/description IdentificationType/description clusterDesc/description clusterDesc/clusterProperties/direction responseModels/steadyStateResponse/propValues/direction objToObjRelation/object/dirIndirSubj TransducerType/identification/ownedBy/email IdentificationType/calibration/validatedBy/email SystemType/identification/operator/email IdentificationType/calibration/calibratedBy/email SystemType/identification/owner/email ProcessType/identification/ownedBy/email IdentificationType/characterization/validatedBy/email IdentificationType/characterization/characterizedBy/email clusterDesc/binHeaderEncode/headerAttrib/encode clusterDesc/dataUnitEncoding/encode clusterDesc/binHeaderEncode/headerAttrib/endian clusterDesc/dataUnitEncoding/endian clusterDesc/binHeaderEncode/headerAttrib/dataUnitFieldSize/endTextDelimiter clusterDesc/dataUnitEncoding/dataUnitFieldSize/endTextDelimiter accuracy/errorDistribution accuracy/factor ValueType/fcnInterpol responseModels/frequencyResponse/freqRespType responseModels/impulseResponse/freqTime clusterDesc/binHeaderEncode/headerAttrib/handleAsType clusterDesc/dataUnitEncoding/handleAsType clusterDesc/binHeaderEncode/headerAttrib/headerAttribName responseModels/steadyStateResponse/responseParameters/hysteresisDirection responseModels/steadyStateResponse/propValues/inputOutput responseModels/steadyStateResponse/dataValues/inputOutput TransducerType/transducerClass/insituRemote

responseModels/steadyStateResponse/responseParameters/proportional
responseModels/steadyStateResponse/propValues/propQualifier
SystemType/relations/propToPropRelation/propUidRef SpatialCoordType/refObjUidRef
objToDataRelation/relationDescription
SystemType/relations/propToPropRelation/relationDescription
objToObjRelation/relationDescription
SystemType/relations/dataToDataRelation/relationDescription
clusterDesc/transSeq/seqOfThisDataStruct clusterDesc/transSeq/sequence
SystemType/identification/serialNumber TransducerType/identification/serialNumber
ProcessType/identification/serialNumber SpatialCoordType/spaceRefSystem
TransducerType/transducerClass/spatialDependancy
cfSubSampling/subSampleCfIndexPts clusterDesc/timeTag/sysClkUidRef
SystemType/relations/timeRelation/sysClkUidRef clusterDesc/idMapping/tapPointUidRef
SystemType/relations/timeRelation/timeCoordinate/timeCoordType
responseModels/steadyStateResponse/responseParameters/timeInvariant
SystemType/relations/timeRelation/timeReference
TransducerType/transducerClass/transmitterReceiver accuracy/type
objToObjRelation/uid objToDataRelation/uid logicalDataStructure/uid
ProcessType/input/inputIdent/uid ProcessType/output/outputIdent/uid dataUnit/uid
SystemType/sysClk/uid SystemType/relations/dataToDataRelation/uid
SystemType/relations/propToPropRelation/uid DataArrayType/uid
DataArrayType/dataSet/uid IdentificationType/uid
responseModels/steadyStateResponse/propValues/UOM
SystemType/relations/dataToDataRelation/dataSource/value objToDataRelation/value
ValueType/valueDataType ValueType/values
responseModels/steadyStateResponse/propValues/variableName dataUnit/variableName
DataArrayType/variableName
responseModels/steadyStateResponse/dataValues/variableName
DataArrayType/dataSet/variableName temporalModel/tcfTrigger/trigType

attributes	Name	Type	Use	Default	Fixed	Annotation
	bindUid	**xs:string**	optional			documentation See BindType
	bindUidRef	**xs:string**	optional			documentation See BindType

annotation	documentation	BindType provides a bindable field that can be used to fill in a value directly, or provide a default value with a way to link it to something else later. When an element contents are variable or when they need to be loaded later (late binding) the bind is used to identify a point in which to attach data later. [The xs:string is used for bindUID attributes throughout. This is used throughout the document for ID, UID and reference attributes.

Example:

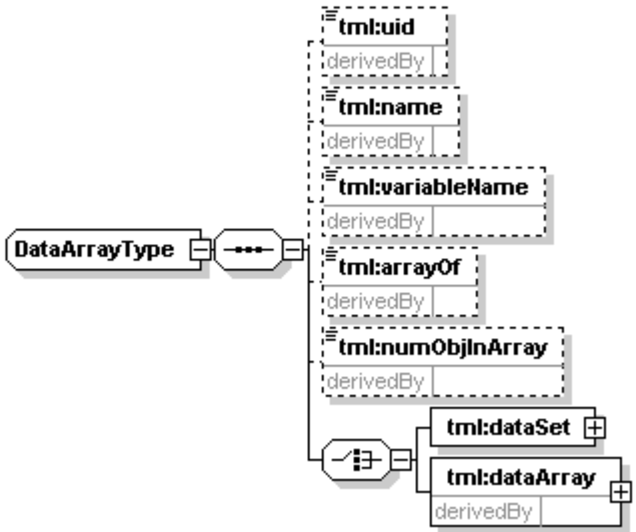
```
<element name="value" type="BindType"/>

<value>5</value>
<value bindUID="SOMEUID"/>
<value bindUID="SOMEUID">5</value>
```

First example you use the value of 5 directly. Second example lookup the changing value elsewhere. Third example you use the default value of 5 and then lookup the changing value elsewhere for the current value.

8.1.1.2.2 complexType DataArrayType

diagram



namespace	http://www.opengis.net/tml							
children	tml:uid tml:name tml:variableName tml:arrayOf tml:numObjInArray tml:dataSet tml:dataArray							
used by	elements	logicalDataStructure/cfDataArray DataArrayType/dataSet/dataArray DataArrayType/dataArray						
attributes	Name	Type	Use	Default	Fixed	Annotation		
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group	

8.1.1.2.2.1 element DataArrayType/uid

diagram

namespace	http://www.opengis.net/tml						
type	tml:BindType						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	xs:string	optional			documentation	See BindType
	bindUidRef	xs:string	optional			documentation	See BindType
annotation	documentation	uid of dataArray					

8.1.1.2.2.2 element DataArrayType/name

diagram

namespace	http://www.opengis.net/tml	
type	tml:BindType	
properties	isRef	0
	minOcc	0
	maxOcc	1

content complex

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation name of dataArray

8.1.1.2.2.3 element DataArrayType/variableName

diagram

namespace http://www.opengis.net/tml

type [**tml:BindType**]

properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation mathematical variable name used in the transformation equations. Index of component is same as order sequence in the LDS.

8.1.1.2.2.4 element DataArrayType/arrayOf

diagram

namespace http://www.opengis.net/tml

type [**tml:BindType**]

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default columns

attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation Allowed values: columns, rows, planes

8.1.1.2.2.5 element DataArrayType/numObjInArray

diagram

namespace http://www.opengis.net/tml

type [**tml:BindType**]

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default 1

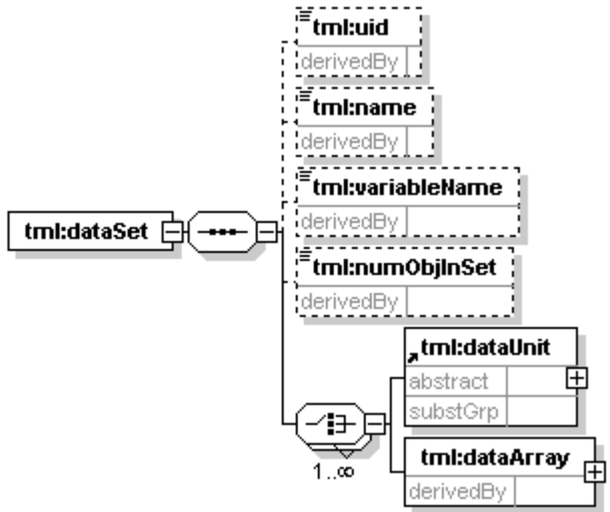
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType

	bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentation: The chosen object (dataSet or dataArray) repeats this many time. default 1

8.1.1.2.2.6 element DataArrayType/dataSet

diagram



namespace	http://www.opengis.net/tml
properties	isRef 0 content complex
children	**tml:uid** **tml:name** **tml:variableName** **tml:numObjInSet** **tml:dataUnit** **tml:dataArray**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation — documentation: data Sets contain a heterogeneous collection of one or more dataUnits

8.1.1.2.2.7 element DataArrayType/dataSet/uid

diagram

namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: uid of dataSet.

8.1.1.2.2.8 element DataArrayType/dataSet/name

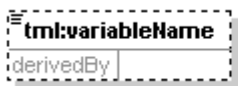
diagram

namespace	http://www.opengis.net/tml
type	**tml:BindType**

properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	name of dataSet					

8.1.1.2.2.9 element DataArrayType/dataSet/variableName

diagram



namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	mathematical variable name used in the transformation equations. Index of component is the order in the sequence in the LDS structure.					

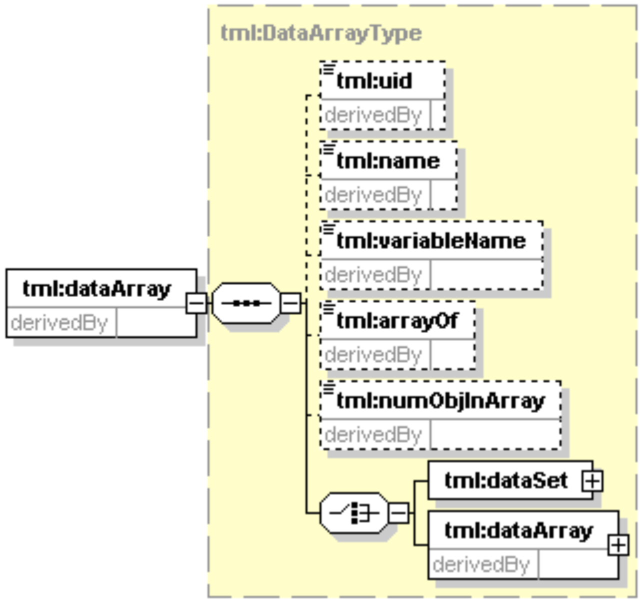
8.1.1.2.2.10 element DataArrayType/dataSet/numObjInSet

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
	default	1					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	number of subordinate sets and/or arrays. default 1					

8.1.1.2.2.11 element DataArrayType/dataSet/dataArray

diagram



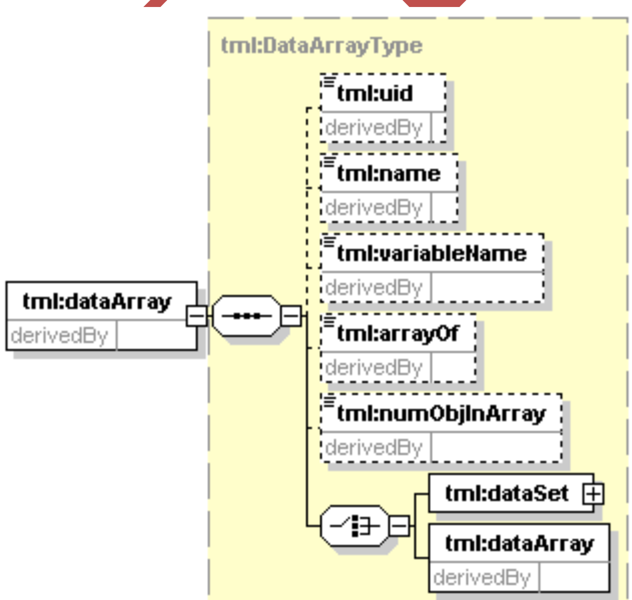
namespace http://www.opengis.net/tml
type tml:DataArrayType
properties isRef 0
content complex
children tml:uid tml:name tml:variableName tml:arrayOf tml:numObjInArray tml:dataSet tml:dataArray
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation dataArrays may be nested

8.1.1.2.2.12 element DataArrayType/dataArray

diagram



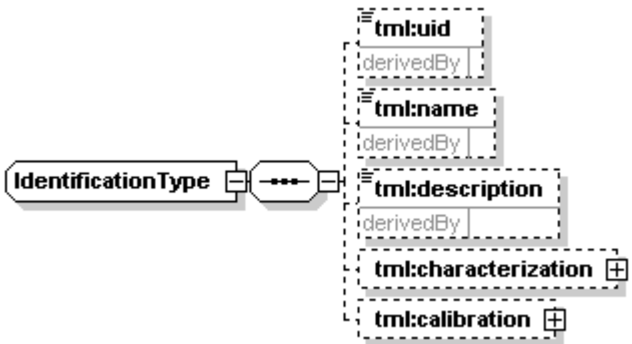
namespace http://www.opengis.net/tml

type **tml:DataArrayType**
properties
isRef 0
content complex
children **tml:uid tml:name tml:variableName tml:arrayOf tml:numObjInArray tml:dataSet tml:dataArray**
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation a dataArray contains a homogeneous collection of one or more dataSets or dataArrays

8.1.1.2.3 complexType IdentificationType

diagram


namespace http://www.opengis.net/tml
children **tml:uid tml:name tml:description tml:characterization tml:calibration**
used by elements **ProcessType/identification SystemType/identification TransducerType/identification**

8.1.1.2.3.1 element IdentificationType/uid

diagram


namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation uid of registry object

8.1.1.2.3.2 element IdentificationType/name

diagram


namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Short descriptive name

8.1.1.2.3.3 element IdentificationType/description

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

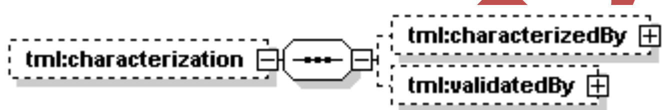
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Longer description

8.1.1.2.3.4 element IdentificationType/characterization

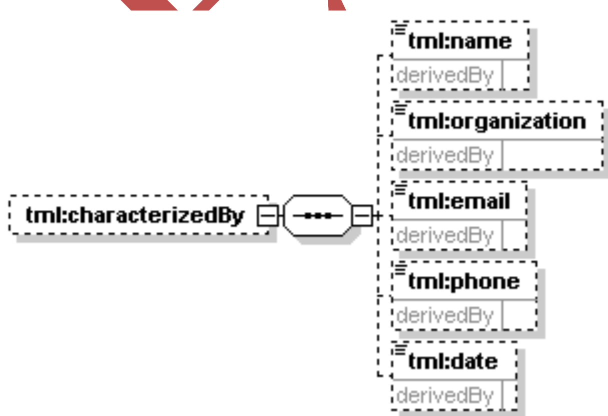
diagram



namespace http://www.opengis.net/tml
properties
isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:characterizedBy** **tml:validatedBy**
annotation documentation Do the tml descriptions comply with the TML Compliance Rules

8.1.1.2.3.5 element IdentificationType/characterization/characterizedBy

diagram



namespace http://www.opengis.net/tml
properties
isRef 0
minOcc 0
maxOcc 1

	content complex
children	tml:name tml:organization tml:email tml:phone tml:date

8.1.1.2.3.6 element IdentificationType/characterization/characterizedBy/name

diagram

namespace	http://www.opengis.net/tml
type	tml:BindType
properties	isRef 0 minOcc 0 maxOcc 1 content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation: documentation Name of person

8.1.1.2.3.7 element IdentificationType/characterization/characterizedBy/organization

diagram

namespace	http://www.opengis.net/tml
type	tml:BindType
properties	isRef 0 minOcc 0 maxOcc 1 content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation: documentation Name of organization

8.1.1.2.3.8 element IdentificationType/characterization/characterizedBy/email

diagram

namespace	http://www.opengis.net/tml
type	tml:BindType
properties	isRef 0 minOcc 0 maxOcc 1 content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation: documentation Email of person

8.1.1.2.3.9 element IdentificationType/characterization/characterizedBy/phone

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation Phone number of person

8.1.1.2.3.10 element IdentificationType/characterization/characterizedBy/date

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

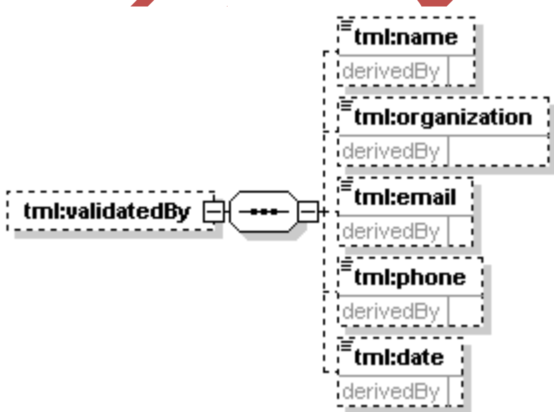
attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation ISO8601 dateTime stamp when characterized

8.1.1.2.3.11 element IdentificationType/characterization/validatedBy

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:name** **tml:organization** **tml:email** **tml:phone** **tml:date**

8.1.1.2.3.12 element IdentificationType/characterization/validatedBy/name

Same as element IdentificationType/characterization/characterizedBy/name

8.1.1.2.3.13 element IdentificationType/characterization/validatedBy/organization

Same as element IdentificationType/characterization/characterizedBy/organization

8.1.1.2.3.14 element IdentificationType/characterization/validatedBy/email

Same as element IdentificationType/characterization/characterizedBy/email

8.1.1.2.3.15 element IdentificationType/characterization/validatedBy/phone

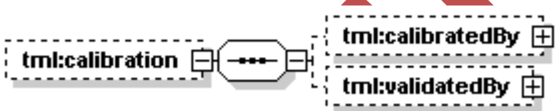
Same as element IdentificationType/characterization/characterizedBy/phone

8.1.1.2.3.16 element IdentificationType/characterization/validatedBy/date

Same as element IdentificationType/characterization/characterizedBy/date

8.1.1.2.3.17 element IdentificationType/calibration

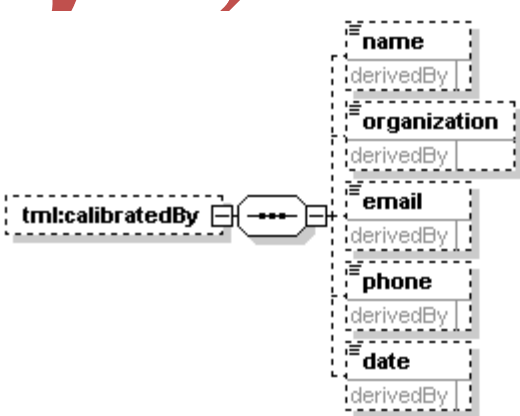
diagram



namespace	http://www.opengis.net/tml		
properties	isRef	0	
	minOcc	0	
	maxOcc	1	
	content	complex	
children	**tml:calibratedBy tml:validatedBy**		
annotation	documentation	Do the TML descriptions accurately reflect actual performance specifications	

8.1.1.2.3.18 element IdentificationType/calibration/calibratedBy

diagram



namespace http://www.opengis.net/tml
properties isRef 0
minOcc 0
maxOcc 1
content complex
children name organization email phone date

8.1.1.2.3.19 element IdentificationType/calibration/calibratedBy/name

Same as element IdentificationType/characterization/characterizedBy/name

8.1.1.2.3.20 element IdentificationType/calibration/calibratedBy/organization

Same as element IdentificationType/characterization/characterizedBy/organization

8.1.1.2.3.21 element IdentificationType/calibration/calibratedBy/email

Same as element IdentificationType/characterization/characterizedBy/email

8.1.1.2.3.22 element IdentificationType/calibration/calibratedBy/phone

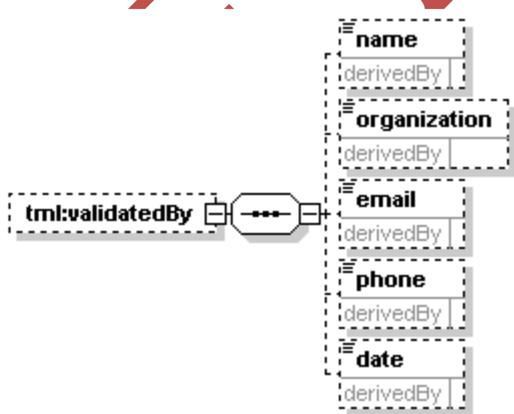
Same as element IdentificationType/characterization/characterizedBy/phone

8.1.1.2.3.23 element IdentificationType/calibration/calibratedBy/date

Same as element IdentificationType/characterization/characterizedBy/date

8.1.1.2.3.24 element IdentificationType/calibration/validatedBy

diagram



namespace http://www.opengis.net/tml
properties isRef 0
minOcc 0
maxOcc 1
content complex
children name organization email phone date

8.1.1.2.3.25 element IdentificationType/calibration/validatedBy/name

Same as element IdentificationType/characterization/characterizedBy/name

8.1.1.2.3.26 element IdentificationType/calibration/validatedBy/organization

Same as element IdentificationType/characterization/characterizedBy/organization

8.1.1.2.3.27 element IdentificationType/calibration/validatedBy/email

Same as element IdentificationType/characterization/characterizedBy/email

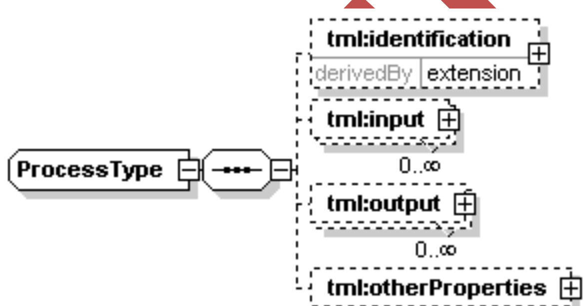
8.1.1.2.3.28 element IdentificationType/calibration/validatedBy/phone

Same as element IdentificationType/characterization/characterizedBy/phone

8.1.1.2.3.29 element IdentificationType/calibration/validatedBy/date

Same as element IdentificationType/characterization/characterizedBy/date

8.1.1.2.4 complexType ProcessType

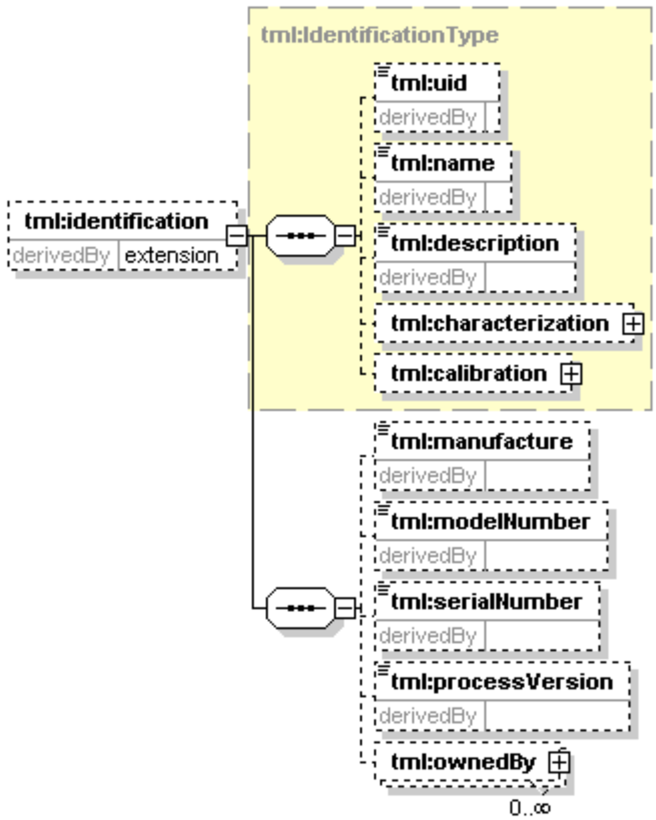
diagram	
namespace	http://www.opengis.net/tml
children	**tml:identification** **tml:input** **tml:output** **tml:otherProperties**
used by	element **process**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group		optional			documentation	See uid_uidRef Attribute Group
SecurityOptionsAttributeGroup		optional			documentation	See ismSecurityOptionsAttribute Group

8.1.1.2.4.1 element ProcessType/identification

diagram



namespace	http://www.opengis.net/tml	
type	extension of **tml:IdentificationType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:uid tml:name tml:description tml:characterization tml:calibration tml:manufacture tml:modelNumber tml:serialNumber tml:processVersion tml:ownedBy**	
annotation	documentation	contains security of process description

8.1.1.2.4.2 element ProcessType/identification/manufacture

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

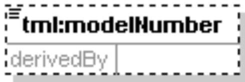
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: Name of Manufacture

8.1.1.2.4.3 element ProcessType/identification/modelNumber

diagram



namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	Model number					

8.1.1.2.4.4 element ProcessType/identification/serialNumber

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	Serial Number					

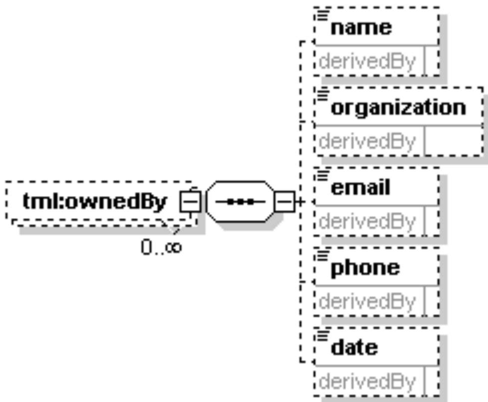
8.1.1.2.4.5 element ProcessType/identification/processVersion

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	Version of software					

8.1.1.2.4.6 element ProcessType/identification/ownedBy

diagram



namespace http://www.opengis.net/tml

properties
- isRef 0
- minOcc 0
- maxOcc unbounded
- content complex

children **name** **organization** **email** **phone** **date**

8.1.1.2.4.7 element ProcessType/identification/ownedBy/name

Same as element IdentificationType/characterization/characterizedBy/name

8.1.1.2.4.8 element ProcessType/identification/ownedBy/organization

Same as element IdentificationType/characterization/characterizedBy/organization

8.1.1.2.4.9 element ProcessType/identification/ownedBy/email

Same as element IdentificationType/characterization/characterizedBy/email

8.1.1.2.4.10 element ProcessType/identification/ownedBy/phone

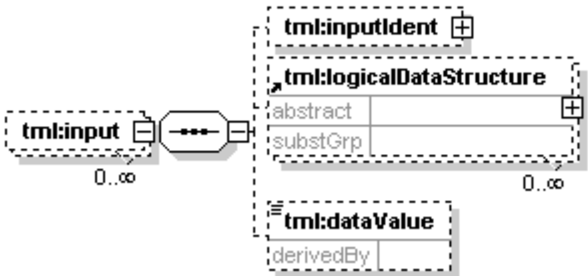
Same as element IdentificationType/characterization/characterizedBy/phone

8.1.1.2.4.11 element ProcessType/identification/ownedBy/date

Same as element IdentificationType/characterization/characterizedBy/date

8.1.1.2.4.12 element ProcessType/input

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children **tml:inputIdent** **tml:logicalDataStructure** **tml:dataValue**

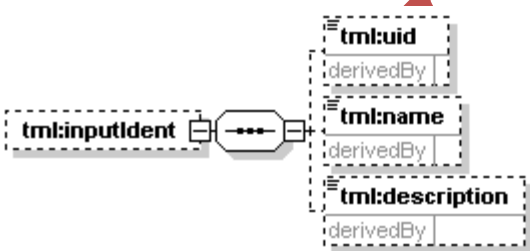
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation a process can have zero or more inputs. This describes a single input process cycle, initiated by an input trigger

8.1.1.2.4.13 element ProcessType/input/inputIdent

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:uid** **tml:name** **tml:description**

8.1.1.2.4.14 element ProcessType/input/inputIdent/uid

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType

	Name	Type	Use	Default	Fixed	Annotation
	bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation uid of input

8.1.1.2.4.15 element ProcessType/input/inputIdent/name

diagram

namespace http://www.opengis.net/tml
type [**tml:BindType**]
properties isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Name of the input

8.1.1.2.4.16 element ProcessType/input/inputIdent/description

diagram

namespace http://www.opengis.net/tml
type [**tml:BindType**]
properties isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Description of the input

8.1.1.2.4.17 element ProcessType/input/dataValue

diagram

namespace http://www.opengis.net/tml
type [**tml:BindType**]
properties isRef 0
minOcc 0
maxOcc 1
content complex

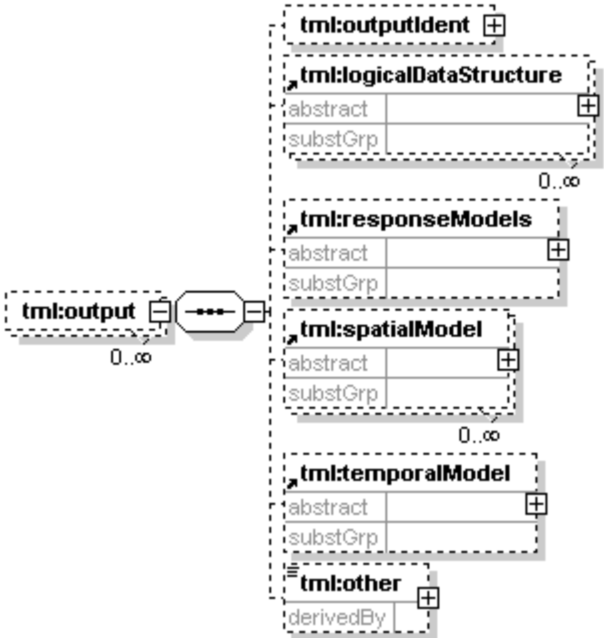
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation fixed or forced input value not. single value or array defined by logical data structure

8.1.1.2.4.18 element ProcessType/output

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children **tml:outputIdent** **tml:logicalDataStructure** **tml:responseModels** **tml:spatialModel** **tml:temporalModel** **tml:other**

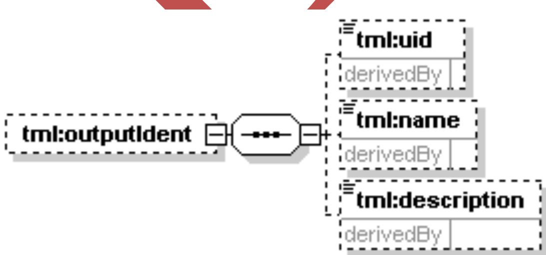
attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation documentation: a process can have one or more outputs. This describes a single output processing cycle, initiated by an output trigger

8.1.1.2.4.19 element ProcessType/output/outputIdent

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:uid** **tml:name** **tml:description**

8.1.1.2.4.20 element ProcessType/output/outputIdent/uid

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	uid of output					

8.1.1.2.4.21 element ProcessType/output/outputIdent/name

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	name of the output					

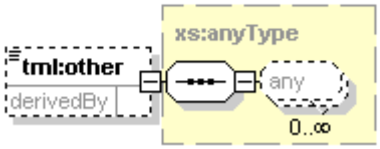
8.1.1.2.4.22 element ProcessType/output/outputIdent/description

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	description of the output					

8.1.1.2.4.23 element ProcessType/output/other

diagram



namespace	http://www.opengis.net/tml	
type	xs:anyType	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
	mixed	true

attributes	Name	Type	Use	Default	Fixed	Annotation

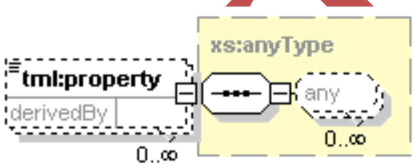
8.1.1.2.4.24 element ProcessType/otherProperties

diagram

namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	tml:property	

8.1.1.2.4.25 element ProcessType/otherProperties/property

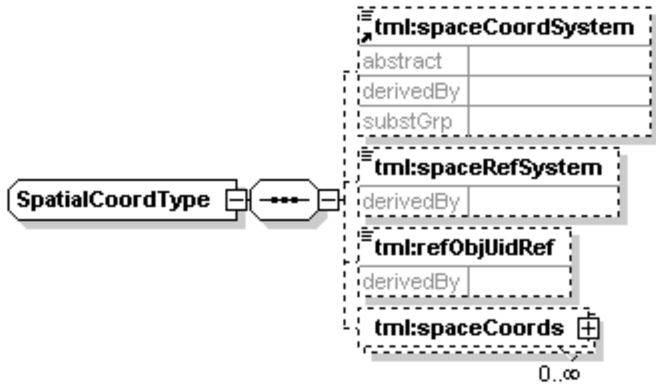
diagram



namespace	http://www.opengis.net/tml	
type	xs:anyType	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
	mixed	true

8.1.1.2.5 complexType SpatialCoordType

diagram



namespace	http://www.opengis.net/tml						
children	**tml:spaceCoordSystem tml:spaceRefSystem tml:refObjUidRef tml:spaceCoords**						
used by	elements	**TransducerType/spatialModel/ambiguitySpace/position SystemType/relations/positionRelation**					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

8.1.1.2.5.1 element SpatialCoordType/spaceRefSystem

diagram

namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	which spatial reference system (i.e. spatial datum) are spatial coordinates referenced (relative) to. Allowed values: transducer, earthCentered, earthLocal, subject. If ref system is transducer or subject then the uid of the transducer or subject must be identified in the refObjUidRef element.					

8.1.1.2.5.2 element SpatialCoordType/refObjUidRef

diagram

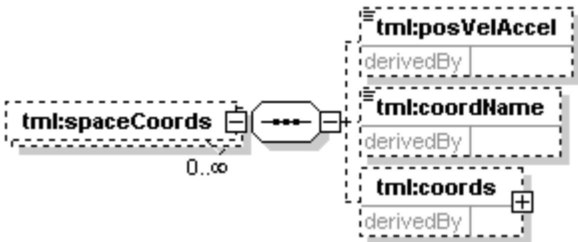
namespace	http://www.opengis.net/tml					
type	**tml:BindType**					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
attributes	Name	Type	Use	Default	Fixed	Annotation

	Name	Type	Use	Default	Fixed	Annotation
	bindUid	**xs:string**	optional			documentation See BindType
	bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: If the spaceRefSystem element is a transducer or a Subject, then this element will identify the particular Transducer or Subject. This is the UID reference of the object which position coordinates are referenced (relative) to.

8.1.1.2.5.3 element SpatialCoordType/spaceCoords

diagram



namespace: http://www.opengis.net/tml

properties:
- isRef 0
- minOcc 0
- maxOcc unbounded
- content complex

children: **tml:posVelAccel** **tml:coordName** **tml:coords**

attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation — documentation: TCF set of positional (translations and rotations) coordinates for each shape, space separated real numbers. Order of coordinates shall be from lowest frequency to highest frequency, same as LDS. Default locations and orientations are zero

8.1.1.2.5.4 element SpatialCoordType/spaceCoords/posVelAccel

diagram

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties:
- isRef 0
- minOcc 0
- maxOcc 1
- content complex
- default pos

attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: Allowed Values: pos, vel, accel, Default is pos.

8.1.1.2.5.5 element SpatialCoordType/spaceCoords/coordName

diagram

namespace: http://www.opengis.net/tml

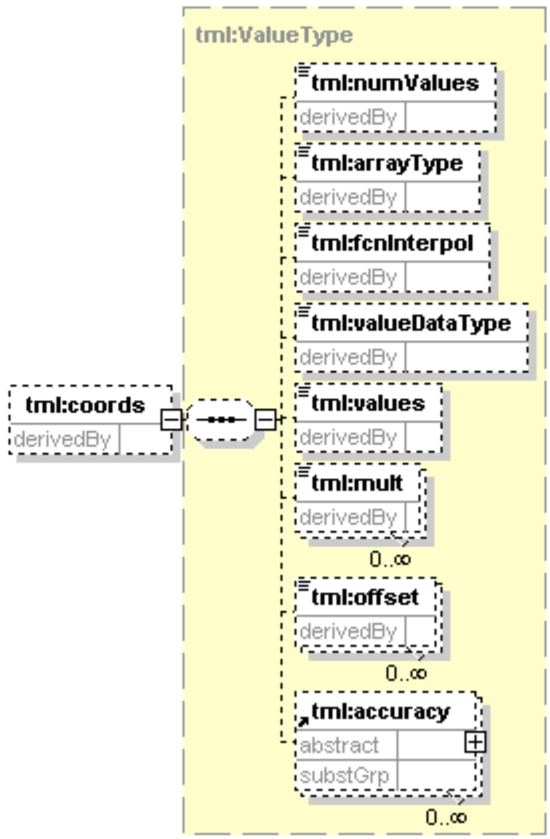
type: **tml:BindType**

properties:
- isRef 0

	minOcc	0					
	maxOcc	1					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	Allowed Values: x, y, z, Alpha, beta, rho, latitude, longitude, altitude, omega, phi, kappa,					

8.1.1.2.5.6 element SpatialCoordType/spaceCoords/coords

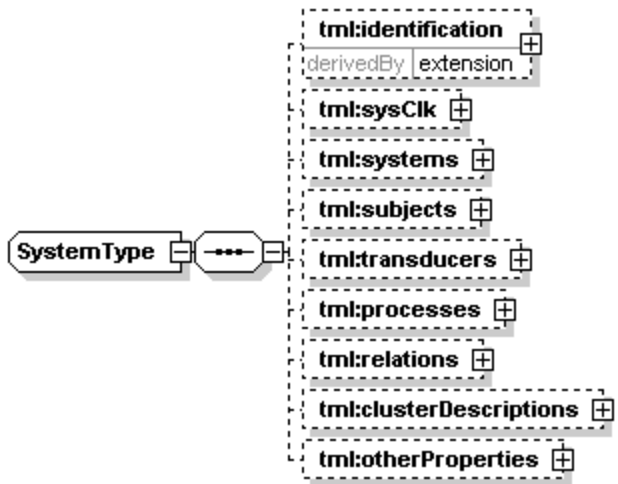
diagram



namespace	http://www.opengis.net/tml						
type	**[tml:ValueType]()**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
children	**tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy**						
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

8.1.1.2.6 complexType SystemType

diagram



namespace http://www.opengis.net/tml

children **tml:identification** **tml:sysClk** **tml:systems** **tml:subjects** **tml:transducers** **tml:processes** **tml:relations** **tml:clusterDescriptions** **tml:otherProperties**

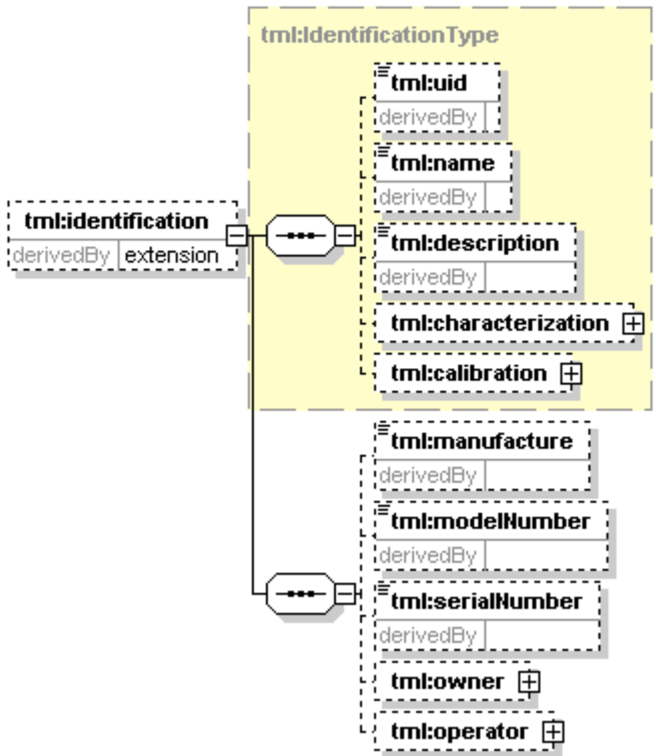
used by element **system**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group		optional			documentation	See uid_uidRef Attribute Group
SecurityOptionsAttributeGroup		optional			documentation	See ismSecurityOptionsAttribute Group

8.1.1.2.6.1 element SystemType/identification

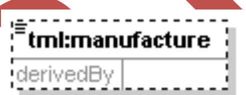
diagram



namespace	http://www.opengis.net/tml	
type	extension of **tml:IdentificationType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:uid tml:name tml:description tml:characterization tml:calibration tml:manufacture tml:modelNumber tml:serialNumber tml:owner tml:operator**	
annotation	documentation	Identification of the system

8.1.1.2.6.2 element SystemType/identification/manufacture

diagram



namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

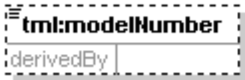
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Name of Manufacture

8.1.1.2.6.3 element SystemType/identification/modelNumber

diagram



namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Model Number

8.1.1.2.6.4 element SystemType/identification/serialNumber

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

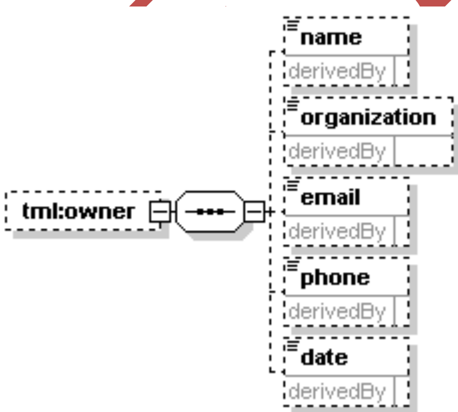
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Serial Number

8.1.1.2.6.5 element SystemType/identification/owner

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **name organization email phone date**

Annotation Child clement details same as child elements under validatedBy and calibratedBy

8.1.1.2.6.6 element SystemType/identification/owner/name

Same as element IdentificationType/characterization/characterizedBy/name

8.1.1.2.6.7 element SystemType/identification/owner/organization

Same as element IdentificationType/characterization/characterizedBy/organization

8.1.1.2.6.8 element SystemType/identification/owner/email

Same as element IdentificationType/characterization/characterizedBy/email

8.1.1.2.6.9 element SystemType/identification/owner/phone

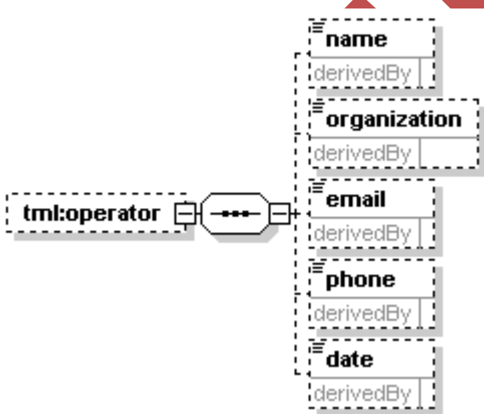
Same as element IdentificationType/characterization/characterizedBy/phone

8.1.1.2.6.10 element SystemType/identification/owner/date

Same as element IdentificationType/characterization/characterizedBy/date

8.1.1.2.6.11 element SystemType/identification/operator

diagram



namespace	http://www.opengis.net/tml
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	**name** **organization** **email** **phone** **date**
annotation	Child clement details same as child elements under validatedBy and calibratedBy

8.1.1.2.6.12 element SystemType/identification/operator /name

Same as element IdentificationType/characterization/characterizedBy/name

8.1.1.2.6.13 element SystemType/identification/operator /organization

Same as element IdentificationType/characterization/characterizedBy/organization

8.1.1.2.6.14 element SystemType/identification/operator /email

Same as element IdentificationType/characterization/characterizedBy/email

8.1.1.2.6.15 element SystemType/identification/operator /phone

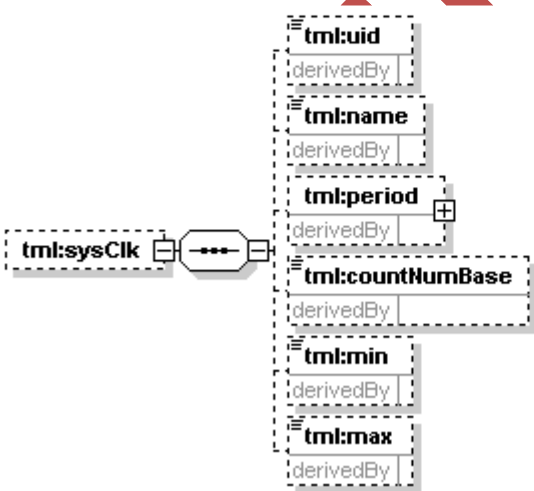
Same as element IdentificationType/characterization/characterizedBy/phone

8.1.1.2.6.16 element SystemType/identification/operator /date

Same as element IdentificationType/characterization/characterizedBy/date

8.1.1.2.6.17 element SystemType/sysClk

diagram



namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	tml:uid tml:name tml:period tml:countNumBase tml:min tml:max	
annotation	documentation	clock counter.

8.1.1.2.6.18 element SystemType/sysClk/uid

diagram 

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation sysClk UID same as the system UID. There is only one clock per system. Subsystems may have clocks

8.1.1.2.6.19 element SystemType/sysClk/name

diagram 

namespace http://www.opengis.net/tml
type **tml:BindType**
properties
isRef 0
minOcc 0
maxOcc 1
content complex

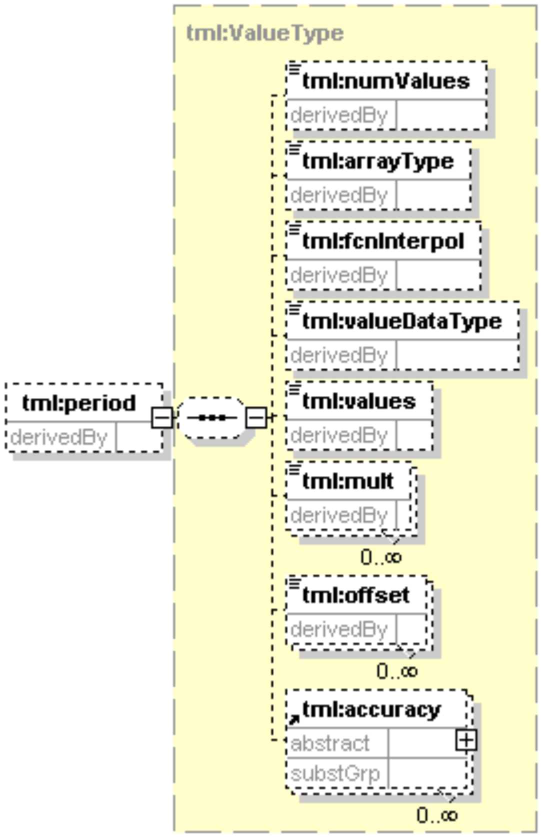
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Name of the system clock

8.1.1.2.6.20 element SystemType/sysClk/period

diagram



namespace	http://www.opengis.net/tml
type	**tml:ValueType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	**tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation — documentation: Period in seconds. The timeTag (clk or dateTime attribute) may have to be calculated after the data. The trigger, which generated the data, may not be available to determine exactly when the data was created, so an approximate time is used

8.1.1.2.6.21 element SystemType/sysClk/countNumBase

diagram

namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1

	content	complex
	default	10

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentation: number base in which clock characters increment. Allowed values are: 2, 8, 10, 16. Default is 10

8.1.1.2.6.22 element SystemType/sysClk/min

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex
default	0

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentatio: counter starting point after rollover. default 0

8.1.1.2.6.23 element SystemType/sysClk/max

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

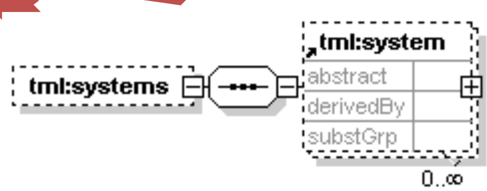
attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentation: max counter count which roll over occurs

8.1.1.2.6.24 element SystemType/systems

diagram



namespace http://www.opengis.net/tml

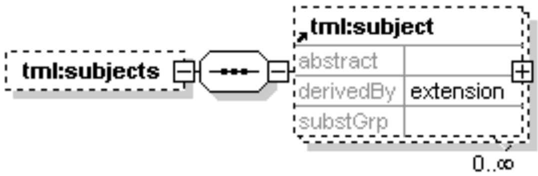
properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:system**

8.1.1.2.6.25 element SystemType/subjects

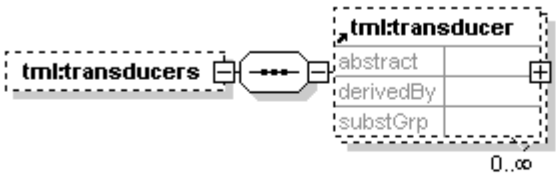
diagram



namespace http://www.opengis.net/tml
properties isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:subject**

8.1.1.2.6.26 element SystemType/transducers

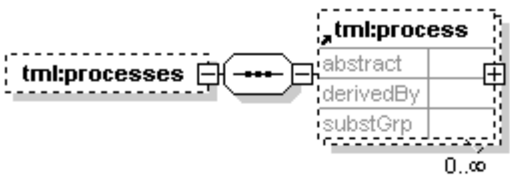
diagram



namespace http://www.opengis.net/tml
properties isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:transducer**

8.1.1.2.6.27 element SystemType/processes

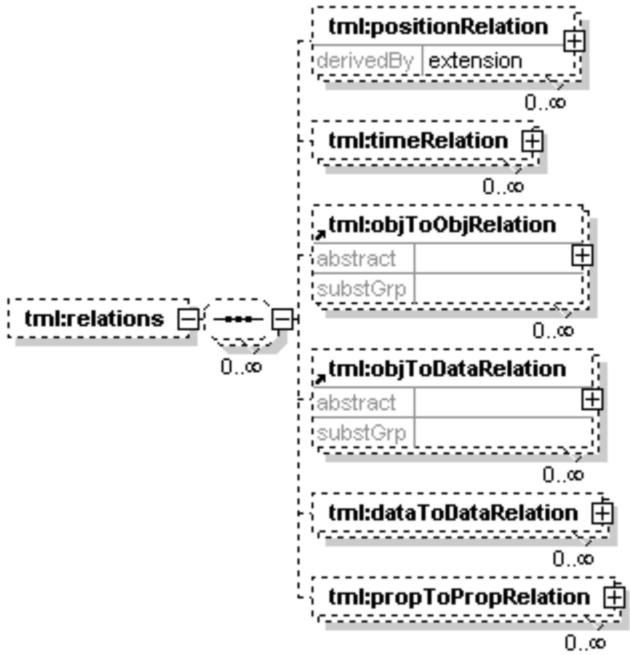
diagram



namespace http://www.opengis.net/tml
properties isRef 0
minOcc 0
maxOcc 1
content complex
children **tml:process**

8.1.1.2.6.28 element SystemType/relations

diagram



namespace http://www.opengis.net/tml

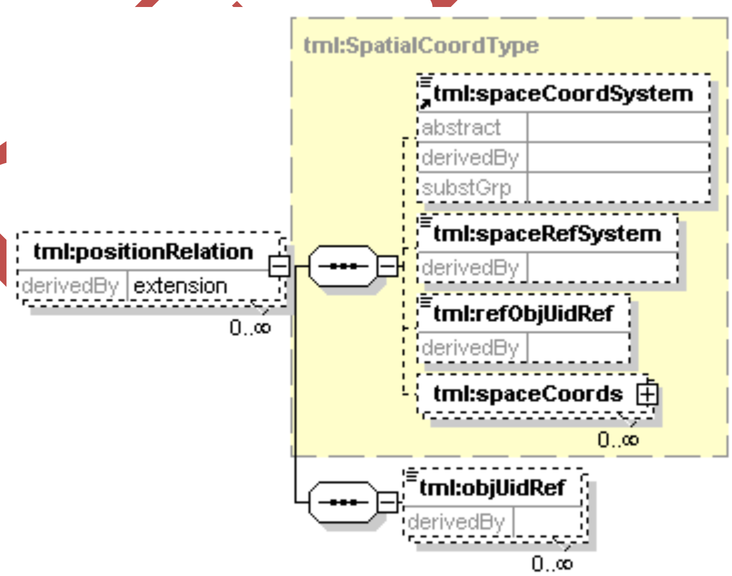
properties
- isRef 0
- minOcc 0
- maxOcc 1
- content complex

children **tml:positionRelation** **tml:timeRelation** **tml:objToObjRelation** **tml:objToDataRelation** **tml:dataToDataRelation** **tml:propToPropRelation**

annotation documentation relationships of objects within the system

8.1.1.2.6.29 element SystemType/relations/positionRelation

diagram



namespace http://www.opengis.net/tml

type: extension of **tml:SpatialCoordType**

properties:

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children: **tml:spaceCoordSystem** **tml:spaceRefSystem** **tml:refObjUidRef** **tml:spaceCoords** **tml:objUidRef**

attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation: documentation: For describing positional relations of subjects external to a system. An empty posRelation tag in a data indicates that this uidRef relation is no longer exist

8.1.1.2.6.30 element SystemType/relations/positionRelation/objUidRef

diagram:

namespace: http://www.opengis.net/tml

type: **tml:BindType**

properties:

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

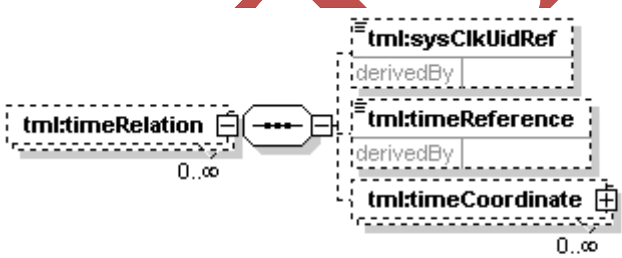
attributes:

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation: See BindType
bindUidRef	**xs:string**	optional			documentation: See BindType

annotation: documentation: uid of the obj being positioned. multiples allowed if in same position and orientation

8.1.1.2.6.31 element SystemType/relations/timeRelation

diagram:



namespace: http://www.opengis.net/tml

properties:

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children: **tml:sysClkUidRef** **tml:timeReference** **tml:timeCoordinate**

attributes:

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation: documentatio: Identifies the absolute time reference for each sysClk. Default is any time reference in a cluster represents absolute time relating to the corresponding clock value. An empty timeRelation tag in a data stream indicates that this uidRef relation is no longer a part of the system

8.1.1.2.6.32 element SystemType/relations/timeRelation/sysClkUidRef

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

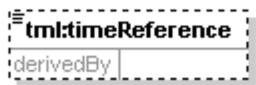
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation UID of the sysClk. Default: Uid of system clock which transducer is contained in.

8.1.1.2.6.33 element SystemType/relations/timeRelation/timeReference

diagram



namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

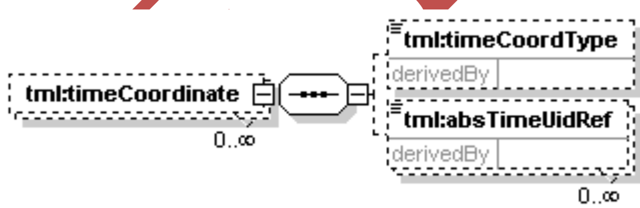
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation time Datum. Allowed Values: UTC, other, Default UTC.

8.1.1.2.6.34 element SystemType/relations/timeRelation/timeCoordinate

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children **tml:timeCoordType** **tml:absTimeUidRef**

8.1.1.2.6.35 element SystemType/relations/timeRelation/timeCoordinate/timeCoordType

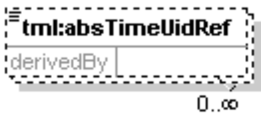
diagram

namespace http://www.opengis.net/tml

type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	1					
	content	complex					
	default	dateTime					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	Allowed values: dateTime, year, mo, day, hour, min, sec. Default: dateTime (ISO 8601)					

8.1.1.2.6.36 element SystemType/relations/timeRelation/timeCoordinate/absTimeUidRef

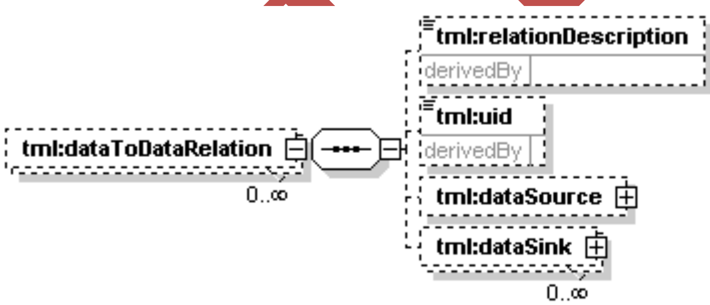
diagram



namespace	http://www.opengis.net/tml						
type	**tml:BindType**						
properties	isRef	0					
	minOcc	0					
	maxOcc	unbounded					
	content	complex					
attributes	Name	Type	Use	Default	Fixed	Annotation	
	bindUid	**xs:string**	optional			documentation	See BindType
	bindUidRef	**xs:string**	optional			documentation	See BindType
annotation	documentation	dataUid reference of the sensor measurements providing the absolute time reference.					

8.1.1.2.6.37 element SystemType/relations/dataToDataRelation

diagram



namespace	http://www.opengis.net/tml						
properties	isRef	0					
	minOcc	0					
	maxOcc	unbounded					
	content	complex					
children	**tml:relationDescription tml:uid tml:dataSource tml:dataSink**						
attributes	Name	Type	Use	Default	Fixed	Annotation	
	uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group
annotation	documentation	Connects bindUIDs to processes. connects outputs to inputs. transducer data to processes and processes to processes. An empty connect tag in a data stream indicates that this UID relation is no longer a part of the system. Example of data to data relation. attaching a process to monitor the state of the gain parameter on the steady state response through a bindUID point.					

8.1.1.2.6.38 element SystemType/relations/dataToDataRelation/relationDescription

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation longer description of the signal or the property relation

8.1.1.2.6.39 element SystemType/relations/dataToDataRelation/uid

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

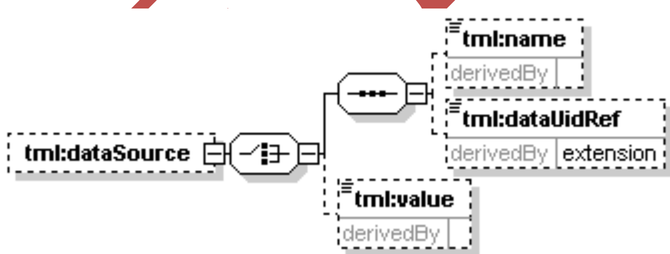
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation connection or node UID of the connection signal data relationship

8.1.1.2.6.40 element SystemType/relations/dataToDataRelation/dataSource

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:name** **tml:dataUidRef** **tml:value**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation documentation data source

8.1.1.2.6.41 element SystemType/relations/dataToDataRelation/dataSource/name

diagram 

namespace http://www.opengis.net/tml
type **tml:BindType**
properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation name of the data source

8.1.1.2.6.42 element SystemType/relations/dataToDataRelation/dataSource/dataUidRef

diagram 

namespace http://www.opengis.net/tml
type extension of **tml:BindType**
properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation UID of the data (live or archived). Archived data streams will have a UID indicative of the data source, time, and clk count of the start.

8.1.1.2.6.43 element SystemType/relations/dataToDataRelation/dataSource/value

diagram 

namespace http://www.opengis.net/tml
type **tml:BindType**
properties

isRef	0
minOcc	0
maxOcc	1
content	complex

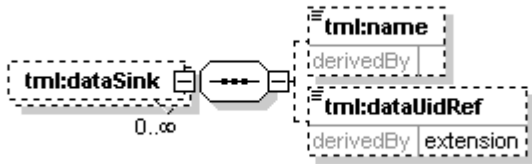
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Direct value for the insertion into data sink

8.1.1.2.6.44 element SystemType/relations/dataToDataRelation/dataSink

diagram



namespace	http://www.opengis.net/tml
properties	isRef 0 minOcc 0 maxOcc unbounded content complex
children	[tml:name]() [tml:dataUidRef]()

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation See uid_uidRef Attribute Group

annotation — documentation: data sink

8.1.1.2.6.45 element SystemType/relations/dataToDataRelation/dataSink/name

diagram

namespace	http://www.opengis.net/tml
type	**[tml:BindType]()**
properties	isRef 0 minOcc 0 maxOcc 1 content complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: Name of data sink

8.1.1.2.6.46 element SystemType/relations/dataToDataRelation/dataSink/dataUidRef

diagram

namespace	http://www.opengis.net/tml
type	extension of **[tml:BindType]()**
properties	isRef 0 minOcc 0 maxOcc 1 content complex

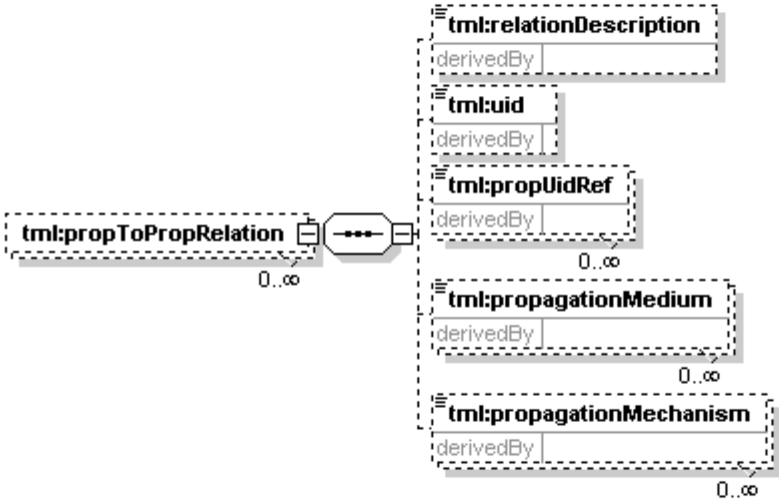
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: UID of the data reference. Archived data streams will have a UID indicative of the data source, time, and clk count of the start.

8.1.1.2.6.47 element SystemType/relations/propToPropRelation

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

children **tml:relationDescription** **tml:uid** **tml:propUidRef** **tml:propagationMedium** **tml:propagationMechanism**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group					documentation	See uid_uidRef Attribute Group

annotation documentation Property to property relation or phenomenon to phenomenon relation. transmitter to receiver, Ambient to receiver, Example: thermal to voltage transducer connected to a voltage to data transducer. example optical filter on the front of an optical camera lens

8.1.1.2.6.48 element SystemType/relations/propToPropRelation/relationDescription

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation longer description of the property relation

8.1.1.2.6.49 element SystemType/relations/propToPropRelation/uid

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation connection or node UID of the property relationship

8.1.1.2.6.50 element SystemType/relations/propToPropRelation/propUidRef

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	unbounded
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation uidRef of the property or phenomenon

8.1.1.2.6.51 element SystemType/relations/propToPropRelation/propagationMedium

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	unbounded
content	complex
default	air

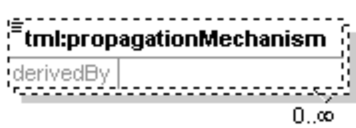
attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation documentation If the Property-to-Property interface has a distance between them, then this describes the medium in which the energy propagates. Allowed values: vacuum, air, water. default air

8.1.1.2.6.52 element SystemType/relations/propToPropRelation/propagationMechanism

diagram



namespace http://www.opengis.net/tml

type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
	default	radiation

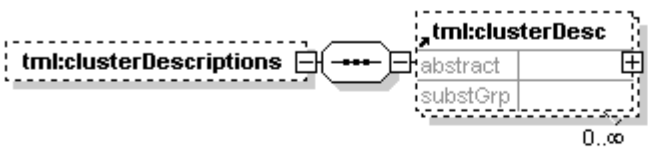
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: If the Property-to-Property interface has a distance between them, then this describes the mechanism in which the energy propagates. Allowed values: radiation, conduction, convection, osmosis. default radiation

8.1.1.2.6.53 element SystemType/clusterDescriptions

diagram



namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:clusterDesc**	

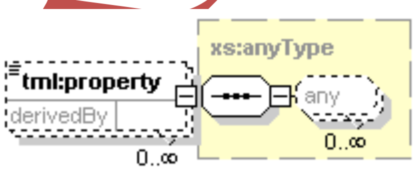
8.1.1.2.6.54 element SystemType/otherProperties

diagram

namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:property**	

8.1.1.2.6.55 element SystemType/otherProperties/property

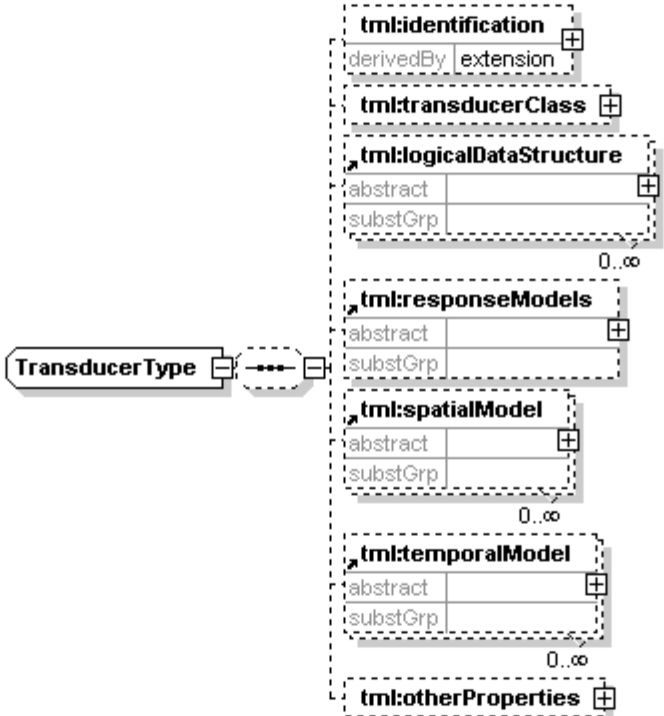
diagram



namespace	http://www.opengis.net/tml	
type	**xs:anyType**	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
	mixed	true

8.1.1.2.7 complexType TransducerType

diagram



namespace http://www.opengis.net/tml

children **tml:identification tml:transducerClass tml:logicalDataStructure tml:responseModels tml:spatialModel tml:temporalModel tml:otherProperties**

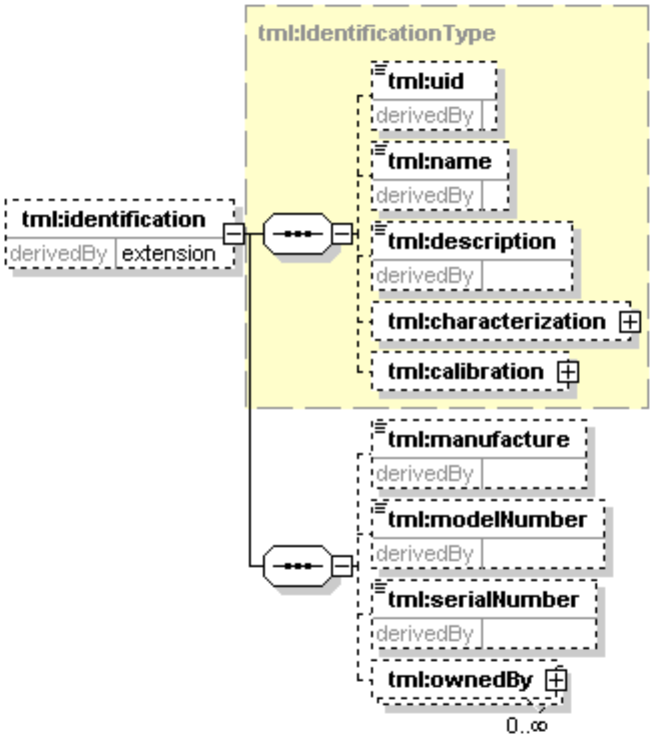
used by element **transducer**

attributes

Name	Type	Use	Default	Fixed	Annotation	
uid_uidRef Attribute Group		optional			documentation	See uid_uidRef Attribute Group
SecurityOptionsAttributeGroup		optional			documentation	See ismSecurityOptionsAttribute Group

8.1.1.2.7.1 element TransducerType/identification

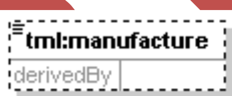
diagram



namespace	http://www.opengis.net/tml
type	extension of **tml:IdentificationType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex
children	**tml:uid tml:name tml:description tml:characterization tml:calibration tml:manufacture tml:modelNumber tml:serialNumber tml:ownedBy**
annotation	documentation bind types on most elements enables the description of transducers in the initialization data stream of data elements.

8.1.1.2.7.2 element TransducerType/identification/manufacture

diagram



namespace	http://www.opengis.net/tml
type	**tml:BindType**
properties	isRef 0 minOcc 0 maxOcc 1 content complex

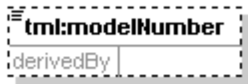
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation — documentation: Name of Manufacture

8.1.1.2.7.3 element TransducerType/identification/modelNumber

diagram



namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Model number

8.1.1.2.7.4 element TransducerType/identification/serialNumber

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

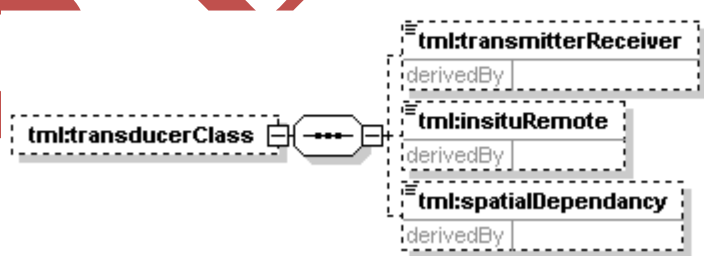
annotation documentation Serial number

8.1.1.2.7.5 element TransducerType/identification/ownedBy

Same as element ProcessType/Identification/ownedBy

8.1.1.2.7.6 element TransducerType/transducerClass

diagram



namespace http://www.opengis.net/tml

properties

isRef	0
minOcc	0
maxOcc	1
content	complex

children **tml:transmitterReceiver** **tml:insituRemote** **tml:spatialDependancy**

annotation documentation Top level transducer classification

8.1.1.2.7.7 element TransducerType/transducerClass/transmitterReceiver

diagram

namespace	http://www.opengis.net/tml					
type	**tml:BindType**					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
	default	receiver				
attributes	Name	Type	Use	Default	Fixed	Annotation
	bindUid	**xs:string**	optional			documentation See BindType
	bindUidRef	**xs:string**	optional			documentation See BindType
annotation	documentation	allowed values: transmitter, receiver, transceiver. default is receiver.				

8.1.1.2.7.8 element TransducerType/transducerClass/insituRemote

diagram

namespace	http://www.opengis.net/tml					
type	**tml:BindType**					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
	default	insitu				
attributes	Name	Type	Use	Default	Fixed	Annotation
	bindUid	**xs:string**	optional			documentation See BindType
	bindUidRef	**xs:string**	optional			documentation See BindType
annotation	documentation	allowed values: insitu, remote. Default is insitu.				

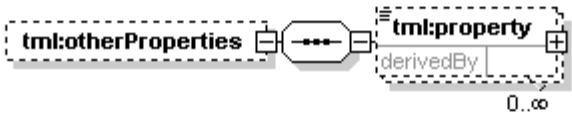
8.1.1.2.7.9 element TransducerType/transducerClass/spatialDependancy

diagram

namespace	http://www.opengis.net/tml					
type	**tml:BindType**					
properties	isRef	0				
	minOcc	0				
	maxOcc	1				
	content	complex				
	default	attitudeIndependent				
attributes	Name	Type	Use	Default	Fixed	Annotation
	bindUid	**xs:string**	optional			documentation See BindType
	bindUidRef	**xs:string**	optional			documentation See BindType
annotation	documentation	Allowed values: attitudeIndependent (default), locationIndependent, positionalIndependent, positionalDependent				

8.1.1.2.7.10 element TransducerType/otherProperties

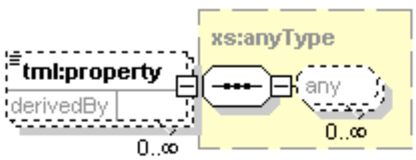
diagram



namespace	http://www.opengis.net/tml	
properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex
children	**tml:property**	

8.1.1.2.7.11 element TransducerType/otherProperties/property

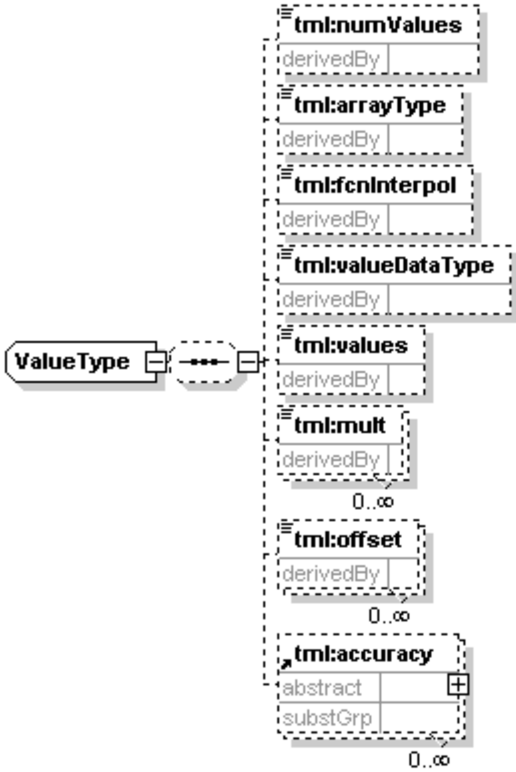
diagram



namespace	http://www.opengis.net/tml	
type	**xs:anyType**	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
	mixed	true

8.1.1.2.8 complexType ValueType

diagram



namespace	http://www.opengis.net/tml	
children	**tml:numValues tml:arrayType tml:fcnInterpol tml:valueDataType tml:values tml:mult tml:offset tml:accuracy**	
used by	elements	**latencyTime TransducerType/temporalModel/ambiguityTime responseModels/impulseResponse/amplitude responseModels/frequencyResponse/amplitude TransducerType/spatialModel/ambiguitySpace/shape/spaceLocCoords/coords SpatialCoordType/spaceCoords/coords responseModels/steadyStateResponse/dataValues responseModels/impulseResponse/frequency responseModels/frequencyResponse/frequency SystemType/sysClk/period TriggerType/period responseModels/frequencyResponse/phase responseModels/steadyStateResponse/propValues TransducerType/temporalModel/tcfDuration TransducerType/temporalModel/tcfOffsetTime responseModels/impulseResponse/time**

attributes

Name	Type	Use	Default	Fixed	Annotation
uid_uidRef Attribute Group					documentation: See uid_uidRef Attribute Group

annotation	documentation	This provides a bindable element for specifying a value along with accuracy information for the value specified.. The value contains a value or a set of values for describing models and functions

8.1.1.2.8.1 element ValueType/numValues

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default 1

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation number of points, or ranges in values element. Allowed values: positive integer. Default is 0.

8.1.1.2.8.2 element ValueType/arrayType

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default singleValue

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed Values: fcn, charFrame. singleValue. Default is fcn. the value element can contain one or multiple values.

8.1.1.2.8.3 element ValueType/fcnInterpol

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties
isRef 0
minOcc 0
maxOcc 1
content complex
default continuous

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation Allowed Values: continuous, discrete, lastValue, returnToZero,

8.1.1.2.8.4 element ValueType/valueDataType

diagram

namespace http://www.opengis.net/tml
type **tml:BindType**

properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation data type of the value. Allowed values: text, number. Default number

8.1.1.2.8.5 element ValueType/values

diagram

namespace http://www.opengis.net/tml

type **tml:BindType**

properties	isRef	0
	minOcc	0
	maxOcc	1
	content	complex

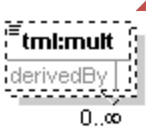
attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation values can contain a single value or a string of values separated by a comma. Each value can contain text, number, or a range of numbers. Each range value shall contain two numbers separated by three decimal points (...), the first number identifies the closed end of the range and the second number identifies the open end of the range. Values in the range may be integer or real numbers. Reals may use E for exponent. In addition to numbers in the range the text -inf and inf can be used to represent minus infinity and plus infinity respectively. For arrayType of function interpolation between values should be handled as indicated in the fcnInterpolate element.

8.1.1.2.8.6 element ValueType/mult

diagram



namespace http://www.opengis.net/tml

type **tml:BindType**

properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
	default	1

attributes

Name	Type	Use	Default	Fixed	Annotation
bindUid	**xs:string**	optional			documentation See BindType
bindUidRef	**xs:string**	optional			documentation See BindType

annotation documentation The multiplier element is applied to the transfer function sequence values. It can be used as a zoom factor for a camera, for example. In a linear function, y=mx+b, mult would be the slope m. If there are multiple values of mult, then each value corresponds to the corresponding Characteristic Frame position or interval. Can have a set of mult or offset equalization values and a sensor modifying those values through a bindUID. The bindUID sensor value will multiply with the values in the mult element. default 1

8.1.1.2.8.7 element ValueType/offset

diagram

namespace	http://www.opengis.net/tml	
type	**tml:BindType**	
properties	isRef	0
	minOcc	0
	maxOcc	unbounded
	content	complex
	default	0

attributes

Name	Type	Use	Default	Fixed	Annotation	
bindUid	**xs:string**	optional			documentation	See BindType
bindUidRef	**xs:string**	optional			documentation	See BindType

annotation — documentation: The offset element is applied to the transfer function sequence values, and is added to the specified sequence value multiplied by the multiplier. In an linear function, y=mx+b, offset would be the intercept b. If there are multiple values of offset, then each value corresponds to the corresponding Characteristic Frame position or interval. Can have a set of mult or offset equalization values and a sensor modifying those values through a bindUID. The bindUID sensor value will add with the values in the offset element. default 0

8.1.1.3 TML Attribute Groups

The following attribute groups are used in TransducerML.

- uid_uidRef

8.1.1.3.1 uid_uidRef Attribute Group

used by

elements	**cfSubSampling clusterDesc dataUnit logicalDataStructure objToDataRelation objToObjRelation responseModels subject tml TransducerType/spatialModel/ambiguitySpace DataArrayType/dataSet SystemType/relations/dataToDataRelation/dataSink SystemType/relations/dataToDataRelation/dataSource SystemType/relations/dataToDataRelation clusterDesc/dataUnitEncoding clusterDesc/binHeaderEncode/headerAttrib ProcessType/input objToObjRelation/object objToDataRelation/object ProcessType/output responseModels/steadyStateResponse/propValues/propName SystemType/relations/propToPropRelation clusterDesc/transSeq/sequence TransducerType/spatialModel/ambiguitySpace/shape SpatialCoordType/spaceCoords TransducerType/spatialModel/ambiguitySpace/shape/spaceLocCoords TransducerType/spatialModel TransducerType/temporalModel SystemType/relations/timeRelation responseModels/steadyStateResponse/propValues/UOM**
complexTypes	**DataArrayType ProcessType SpatialCoordType SystemType TransducerType TriggerType ValueType**

attributes

Name	Type	Use	Default	Fixed	Annotation	
name	**xs:string**	optional			documentation	short descriptive name of element
uid	**xs:anyURI**	optional			docu	unique ID for this element

			ment ation	
uidRef	**xs:anyURI**	optional	documentation	the contents of this element are exactly the same as the contents of the uidRef element. no need repeating it. (similar to xlink) it. (similar to xlink)

8.1.2. ism Elements

The following is a list of element, complex types and attributes groups described in this section.

Simple types	Attr. groups
ClassificationType	**SecurityAttributesOptionGroup**

8.1.2.1 ism Simple Elements

simpleType **ism:ClassificationType**

namespace	urn:us:gov:ic:ism:v2	
type	restriction of **xs:NMTOKEN**	
used by	attribute	**data/@ismclass**
facets	enumeration	U
	enumeration	C
	enumeration	S
	enumeration	TS
	enumeration	R
	enumeration	CTS
	enumeration	CTS-B
	enumeration	CTS-BALK
	enumeration	NU
	enumeration	NR
	enumeration	NC
	enumeration	NS
	enumeration	NS-S
	enumeration	NS-A
	enumeration	CTSA
	enumeration	NSAT
	enumeration	NCA
annotation	documentation	A simple type in which a name token is restricted to the US, non-US, and joint classification portion mark abbreviations from the CAPCO Register.

8.1.2.2 ism:Attribute Groups

The following attribute groups are used in TransducerML from the urn:us:gov:ic:ism:v2 namespace

- ismSecurityOptionsAttributeGroup

8.1.2.2.1 ism:SecurityAttributesOptionGroup

namespace	urn:us:gov:ic:ism:v2					
used by	elements	**clusterDesc objToDataRelation objToObjRelation tml**				
	complexTypes	**ProcessType SystemType TransducerType**				
attributes	Name	Type	Use	Default	Fixed	Annotation

classification	optional		
ownerProducer	optional	documentation	ISO 3166-1 trigraph(s) of the owner or producer country(ies) and/or CAPCO-specified tetragraphs of international organizations. Either (a) a single trigraph or tetragraph or (b) a space-delimited list of trigraphs followed by tetragraphs. Trigraphs must be in alphabetical order and tetragraphs must be in alphabetical order.
SCIcontrols	optional	documentation	Authorized abbreviation(s) of SCI control system(s). Either (a) a single abbreviation or (b) a space-delimited list of abbreviations in the order prescribed in the CAPCO Register.
SARIdentifier	optional	documentation	Authorized Special Access Required program digraph(s) or trigraph(s) preceded by "SAR-". Either (a) a single digraph or trigraph or (b) a space-delimited list of digraphs or trigraphs. Example: "SAR-ABC SAR-DEF ..."
disseminationControls	optional	documentation	Authorized dissemination control portion mark abbreviation(s). Either (a) a single abbreviation or (b) a space-delimited list of abbreviations in the order shown in the CAPCO Register. Exception: For the "REL" abbreviation, omit the country code trigraph(s) and instead place the trigraph(s) in the "releasableTo" attribute value.
FGIsourceOpen	optional	documentation	Non-US classification portion marking for foreign government information in a document portion. Use this attribute to record a source country when the intent is to post the document to a shared space with the source identified.
FGIsourceProtected	optional	documentation	Non-US classification portion marking for foreign government information in a document portion. Use this attribute to record a source country when the intent is to filter out the identity of the source prior to posting the document to a shared space.
releasableTo	optional	documentation	ISO 3166-1 trigraphic codes of countries to which the associated content can be released. Include "USA" in all instances. Use a space- delimited list with "USA" first, followed by the other trigraph(s) in alphabetical order.
nonICmarkings	optional	documentation	Authorized non-IC portion marking abbreviation(s) from the CAPCO Register. Either (a) a single non-IC abbreviated marking or (b) a space-delimited list of abbreviations in the order shown in the CAPCO Register.

classifiedBy	optional	documentation	Use as specified by E.O. 12958.
classificationReason	optional	documentation	A text string containing one or more paragraph numbers, 1.4(a) through 1.4(h), taken from E.O. 12958, as amended. Enter the paragraph references as they should appear in a classification/declassification block.
derivedFrom	optional	documentation	A text string containing (a) the title and date of a specific source document, or (b) the title and date of an organization classification guide, or (c) the literal string "Multiple Sources".
declassDate	optional	documentation	A specific date, in the format YYYY-MM-DD, at which the applicable information is automatically declassified.
declassEvent	optional	documentation	A textual description of an event that triggers declassification.
declassException	optional	documentation	One or more of the exceptions to 25-year declassification: specify "25X1-human", "25X1", "25X2", ..., "25X9". If more than one exception applies, use a space-delimited list. If "25X1-human" applies, it should be first in a list.
typeOfExemptedSource	optional	documentation	One or more tokens indicating that a source that is exempted from automatic declassification applies. Specify "OADR", "X1", "X2", ..., "X8". If more than one applies, use a space-delimited list.
dateOfExemptedSource	optional	documentation	A specific date, in the format YYYY-MM-DD. Used in conjunction with attribute "typeOfExemptedSource." If there are multiple exempted sources, specify the date of the exempted source that has the most recent date.
declassManualReview	optional	documentation	A true/false indication that manual review is required for declassification. Use this attribute to force the appearance of "//MR" in the header and footer marking titles. Use this attribute ONLY when it is necessary to override the business logic applied to classification and control markings in the document to determine whether manual review is required.

annotation	documentation	The group of Information Security Marking attributes in which the use of attributes 'classification' and 'ownerProducer' is optional. This group is to be contrasted with group 'SecurityAttributesGroup' in which use of these attributes is required.

Annex A
(Normative)

XML Schema Documents

In addition to this document, this specification includes several normative XML Schema Documents. These XML Schema Documents for Version 1.0.0, are posted online at the URL http://schemas.opengis.net/tml/1.0.0/. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online version shall be considered authoritative.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tml="http://www.opengis.net/tml"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ism="urn:us:gov:ic:ism:v2"
targetNamespace="http://www.opengis.net/tml" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:import namespace="urn:us:gov:ic:ism:v2" schemaLocation="IC-ISM-v2.xsd"/>
  <xs:element name="tml">
    <xs:annotation>
      <xs:documentation>Root Element.  Also contains attributes to indicate the
overall security classification of this TML stream or file.  If needed individual
data clusters can be labeled with a security class.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element ref="tml:system"/>
        <xs:element ref="tml:subject"/>
        <xs:element ref="tml:transducer"/>
        <xs:element ref="tml:process"/>
        <xs:element name="extSysRelations">
          <xs:annotation>
            <xs:documentation>for relating external subject to external  subject or
transducer data to external subject.  An external subject (object) is external to
the system.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="tml:objToObjRelation" minOccurs="0"
maxOccurs="unbounded"/>
              <xs:element ref="tml:objToDataRelation" minOccurs="0"
maxOccurs="unbounded"/>
              <xs:element name="otherRelations" type="xs:anyType" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element ref="tml:data"/>
      </xs:choice>
      <xs:attributeGroup ref="tml:uid_uidRef"/>
      <xs:attribute name="version" type="xs:string" use="required" fixed="1.0">
        <xs:annotation>
          <xs:documentation>fixed version 1.0</xs:documentation>
```

```
      </xs:annotation>
     </xs:attribute>
     <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
   </xs:complexType>
  </xs:element>
  <!--Base Types-->
  <xs:complexType name="BindType">
   <xs:simpleContent>
     <xs:extension base="xs:string">
      <xs:attribute name="bindUid" type="xs:string" use="optional"/>
      <xs:attribute name="bindUidRef" type="xs:string" use="optional"/>
     </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="ValueType">
   <xs:annotation>
     <xs:documentation>contains a value or a set of values for describing models
and functions</xs:documentation>
   </xs:annotation>
   <xs:sequence minOccurs="0">
     <xs:element name="numValues" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>number of points, or ranges in values element.   Allowed
values: positive integer. Default is 0.</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="arrayType" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Allowed Values: fcn, charFrame. singleValue.  Default is
fcn.   the value element can contain one or multiple values. </xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="fcnInterpol" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Allowed Values: continuous, discrete, lastValue,
returnToZero, </xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="valueDataType" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>data type of the value. Allowed values: text, number.
Default number</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="values" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>values can contain a single value or a string of values
separated by a comma.  Each value can contain text,  number, or a range of
numbers. Each range value shall contain two numbers separated by three decimal
points (...), the first number identifies the closed end of the range and the
second number identifies the open end of the range.  Values in the range may be
integer or real numbers. Reals may use E for exponent.   In addition to numbers in
the range the text -inf and inf can be used to represent -infinity and plus
infinity respectively.  For arrayType of function interpolation between values
should be handled as indicated in the fcnInterpolate element.</xs:documentation>
      </xs:annotation>
```

```
    </xs:element>
    <xs:element name="mult" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>default 1. if multiple values of mult, then each value
corresponds to the corresponding Characteristic Frame position or interval. Can
have a set of mult or offset equalization values and a sensor modifying those
values through a bindUID.  The bindUID sensor value will multiply with the values
in the mult element.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="offset" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>default 0. if multiple values of offset, then each value
corresponds to the corresponding Characteristic Frame  position or interval.  Can
have a set of mult or offset equalization values and a sensor modifying those
values through a bindUID.  The bindUID sensor value will add with the values in
the offset element.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="tml:accuracy" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attributeGroup ref="tml:uid_uidRef"/>
  </xs:complexType>
  <!--Element Types-->
  <xs:complexType name="ProcessType">
   <xs:sequence>
    <xs:element name="identification" minOccurs="0">
      <xs:annotation>
        <xs:documentation>contains security of process
description</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="tml:IdentificationType">
            <xs:sequence>
              <xs:element name="manufacture" type="tml:BindType" minOccurs="0"/>
              <xs:element name="modelNumber" type="tml:BindType" minOccurs="0"/>
              <xs:element name="serialNumber" type="tml:BindType" minOccurs="0"/>
              <xs:element name="processVersion" type="tml:BindType" minOccurs="0"/>
              <xs:element name="ownedBy" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                    <xs:element name="organization" type="tml:BindType"
minOccurs="0"/>
                    <xs:element name="email" type="tml:BindType" minOccurs="0"/>
                    <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
                    <xs:element name="date" type="tml:BindType" minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
```



```
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:element>
<xs:element name="input" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>a process can have zero or more inputs. This describes a
single input process cycle, initiated by an input trigger</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:annotation>
        <xs:documentation>  </xs:documentation>
      </xs:annotation>
      <xs:element name="inputIdent" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="uid" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>uid of input</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="name" type="tml:BindType" minOccurs="0"/>
            <xs:element name="description" type="tml:BindType" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="tml:logicalDataStructure" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="dataValue" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>fixed or forced input value not.  single value or
array defined by logical data structure </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="tml:uid_uidRef"/>
  </xs:complexType>
</xs:element>
<xs:element name="output" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>a process can have one or more outputs.  This describes a
single output processing cycle, initiated by an output trigger </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="outputIdent" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="uid" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>uid of output</xs:documentation>
              </xs:annotation>
            </xs:element>
```

```
                <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                <xs:element name="description" type="tml:BindType" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element ref="tml:logicalDataStructure" minOccurs="0"
maxOccurs="unbounded"/>
          <xs:element ref="tml:responseModels" minOccurs="0"/>
          <xs:element ref="tml:spatialModel" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="tml:temporalModel" minOccurs="0"/>
          <xs:element name="other" type="xs:anyType" minOccurs="0"/>
        </xs:sequence>
        <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="otherProperties" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="property" type="xs:anyType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
  <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
</xs:complexType>
<xs:complexType name="SystemType">
  <xs:sequence>
    <xs:element name="identification" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Identification of the system</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="tml:IdentificationType">
            <xs:sequence>
              <xs:element name="manufacture" type="tml:BindType" minOccurs="0"/>
              <xs:element name="modelNumber" type="tml:BindType" minOccurs="0"/>
              <xs:element name="serialNumber" type="tml:BindType" minOccurs="0"/>
              <xs:element name="owner" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                    <xs:element name="organization" type="tml:BindType"
minOccurs="0"/>
                    <xs:element name="email" type="tml:BindType" minOccurs="0"/>
                    <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
                    <xs:element name="date" type="tml:BindType" minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
```



```
          <xs:element name="operator" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                <xs:element name="organization" type="tml:BindType"
minOccurs="0"/>
                <xs:element name="email" type="tml:BindType" minOccurs="0"/>
                <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
                <xs:element name="date" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
<xs:element name="sysClk" minOccurs="0">
  <xs:annotation>
    <xs:documentation>clock counter.  </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="uid" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>sysClk UID same as the system UID.  There is only
one clock per system.  Subsystems may have clocks</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="name" type="tml:BindType" minOccurs="0"/>
      <xs:element name="period" type="tml:ValueType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Period in seconds</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="countNumBase" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>number base in which clock characters increment.
Allowed values are: 2, 8, 10, 16.  Default is 10</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="min" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>counter starting point after rollover.  default
0</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="max" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>max counter count which roll over
occurs</xs:documentation>
        </xs:annotation>
```

```
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="systems" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="tml:system" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="subjects" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="tml:subject" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="transducers" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="tml:transducer" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="processes" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="tml:process" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="relations" minOccurs="0">
      <xs:annotation>
        <xs:documentation>relationships of objects within the system.
characterized at the time of the system characterization.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
          <xs:element name="positionRelation" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>For describing positional relations of subjects
external to a system.  An empty posRelation tag in a data indicates that this
uidRef relation is no longer exist</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="tml:SpatialCoordType">
                  <xs:sequence>
                    <xs:element name="objUidRef" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
                      <xs:annotation>
                        <xs:documentation>uid of the obj being positioned. multiples
allowed if in same position and orientation</xs:documentation>
                      </xs:annotation>
                    </xs:element>
```

```
          </xs:sequence>
         </xs:extension>
        </xs:complexContent>
       </xs:complexType>
      </xs:element>
      <xs:element name="timeRelation" minOccurs="0" maxOccurs="unbounded">
       <xs:annotation>
        <xs:documentation>Identifies the absolute time reference for each
sysClk.  Default is any time reference in a cluster represents absolute time
relating to the corresponding clock value.  An empty timeRelation tag in a data
stream indicates that this uidRef relation is no longer a part of the
system</xs:documentation>
       </xs:annotation>
       <xs:complexType>
        <xs:sequence>
         <xs:element name="sysClkUidRef" type="tml:BindType" minOccurs="0">
          <xs:annotation>
           <xs:documentation>UID of the sysClk.  Default: Uid of system clock
which transducer is contained in.</xs:documentation>
          </xs:annotation>
         </xs:element>
         <xs:element name="timeReference" type="tml:BindType" minOccurs="0">
          <xs:annotation>
           <xs:documentation>time Datum.  Allowed Values: UTC, other,
Default UTC.</xs:documentation>
          </xs:annotation>
         </xs:element>
         <xs:element name="timeCoordinate" minOccurs="0"
maxOccurs="unbounded">
          <xs:complexType>
           <xs:sequence>
            <xs:element name="timeCoordType" type="tml:BindType"
minOccurs="0">
             <xs:annotation>
              <xs:documentation>Allowed values: dateTime,  year, mo, day,
hour, min, sec. Default: dateTime (ISO 8601)</xs:documentation>
             </xs:annotation>
            </xs:element>
            <xs:element name="absTimeUidRef" type="tml:BindType"
minOccurs="0" maxOccurs="unbounded">
             <xs:annotation>
              <xs:documentation>dataUid reference of the sensor
measurements providing the absolute time reference.</xs:documentation>
             </xs:annotation>
            </xs:element>
           </xs:sequence>
          </xs:complexType>
         </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="tml:uid_uidRef"/>
       </xs:complexType>
      </xs:element>
      <xs:element ref="tml:objToObjRelation" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="tml:objToDataRelation" minOccurs="0"
maxOccurs="unbounded"/>
```

```
        <xs:element name="dataToDataRelation" minOccurs="0"
maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Connects bindUIDs to processes. connects outputs to
inputs. transducer data to processes and processes to processes.  An empty connect
tag in a data stream indicates that this UID relation is no longer a part of the
system. Example of data to data relation.  attaching a process to monitor the
state of the gain parameter on the steady state response through a bindUID point.
</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="relationDescription" type="tml:BindType"
minOccurs="0">
                <xs:annotation>
                  <xs:documentation>longer description of the signal or the property
relation</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="uid" type="tml:BindType" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>connection or node UID of the connection signal
data relationship</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="dataSource" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>data source</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:choice>
                    <xs:sequence>
                      <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                      <xs:element name="dataUidRef" minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>UID of the data (live or archived).
Archived data streams will have a UID indicative of the data source, time, and clk
count of the start. </xs:documentation>
                        </xs:annotation>
                        <xs:complexType>
                          <xs:simpleContent>
                            <xs:extension base="tml:BindType"/>
                          </xs:simpleContent>
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                    <xs:element name="value" type="tml:BindType" minOccurs="0"/>
                  </xs:choice>
                  <xs:attributeGroup ref="tml:uid_uidRef"/>
                </xs:complexType>
              </xs:element>
              <xs:element name="dataSink" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                  <xs:documentation>data sink</xs:documentation>
                </xs:annotation>
                <xs:complexType>
```



```
                <xs:sequence>
                  <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                  <xs:element name="dataUidRef" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>UID of the data reference.  Archived data
streams will have a UID indicative of the data source, time, and clk count of the
start. </xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="tml:BindType"/>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attributeGroup ref="tml:uid_uidRef"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attributeGroup ref="tml:uid_uidRef"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="propToPropRelation" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Property to property relation or phenomenon to
phenomenon relation. transmitter to receiver, Ambient to receiver, Example:
thermal to voltage transducer connected to a voltage to data transducer.  example
optical filter on the front of an optical camera lens</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="relationDescription" type="tml:BindType"
minOccurs="0">
              <xs:annotation>
                <xs:documentation>longer description of  the property
relation</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="uid" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>connection or node UID of the property
relationship</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="propUidRef" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>uidRef of the property or
phenomenon</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="propagationMedium" type="tml:BindType"
minOccurs="0" maxOccurs="unbounded">
              <xs:annotation>
```

```
                <xs:documentation>If the P-to-P interface has a distance between
them, then this describes the medium in which the energy propagates.  Allowed
values: vacuum, air, water.  default air</xs:documentation>
               </xs:annotation>
              </xs:element>
              <xs:element name="propagationMechanism" type="tml:BindType"
minOccurs="0" maxOccurs="unbounded">
               <xs:annotation>
                <xs:documentation>If the P-to-P interface has a distance between
them, then this describes the mechanism in which the energy propagates.  Allowed
values: radiation, conduction, convection, osmosis.  default
radiation</xs:documentation>
               </xs:annotation>
              </xs:element>
             </xs:sequence>
             <xs:attributeGroup ref="tml:uid_uidRef"/>
            </xs:complexType>
           </xs:element>
          </xs:sequence>
         </xs:complexType>
        </xs:element>
        <xs:element name="clusterDescriptions" minOccurs="0">
         <xs:complexType>
          <xs:sequence>
           <xs:element ref="tml:clusterDesc" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
         </xs:complexType>
        </xs:element>
        <xs:element name="otherProperties" minOccurs="0">
         <xs:complexType>
          <xs:sequence>
           <xs:element name="property" type="xs:anyType" minOccurs="0"
maxOccurs="unbounded"/>
          </xs:sequence>
         </xs:complexType>
        </xs:element>
       </xs:sequence>
       <xs:attributeGroup ref="tml:uid_uidRef"/>
       <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
      </xs:complexType>
      <xs:complexType name="TransducerType">
       <xs:sequence minOccurs="0">
        <xs:element name="identification" minOccurs="0">
         <xs:annotation>
          <xs:documentation>bind types on most elements enables the description of
transducers in the initialization data stream of data elements.
</xs:documentation>
         </xs:annotation>
         <xs:complexType>
          <xs:complexContent>
           <xs:extension base="tml:IdentificationType">
            <xs:sequence>
             <xs:element name="manufacture" type="tml:BindType" minOccurs="0"/>
             <xs:element name="modelNumber" type="tml:BindType" minOccurs="0"/>
             <xs:element name="serialNumber" type="tml:BindType" minOccurs="0"/>
             <xs:element name="ownedBy" minOccurs="0" maxOccurs="unbounded">
```



```
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                      <xs:element name="organization" type="tml:BindType"
minOccurs="0"/>
                      <xs:element name="email" type="tml:BindType" minOccurs="0"/>
                      <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
                      <xs:element name="date" type="tml:BindType" minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="transducerClass" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Top level transducer classification</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="transmitterReceiver" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>allowed values: transmitter, receiver, transceiver.
default is receiver.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="insituRemote" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>allowed values: insitu, remote.  Default is
insitu.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="spatialDependancy" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Allowed values: attitudeIndependent (default),
locationIndependent, positionalIndependent, positionalDependent</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element ref="tml:logicalDataStructure" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="tml:responseModels" minOccurs="0"/>
      <xs:element ref="tml:spatialModel" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="tml:temporalModel" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="otherProperties" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
```



```
          <xs:element name="property" type="xs:anyType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
  <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
</xs:complexType>
<!--Support Types-->
<xs:complexType name="DataArrayType">
  <xs:sequence>
    <xs:element name="uid" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>uid of dataArray</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="name" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>name of dataArray</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="variableName" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Name of mathematical term used in the transformation
equations.  Index of component is same as order sequence in the
lds.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="arrayOf" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Allowed values: columns, rows, planes default is
columns</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="numObjInArray" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The chosen object (dataSet or dataArray) repeats this
many time.   default 1</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:choice minOccurs="0">
      <xs:annotation>
        <xs:documentation>a dataArrays contain a homogeneous collection of either
subordinate dataArrays or dataSets. </xs:documentation>
      </xs:annotation>
      <xs:element name="dataSet">
        <xs:annotation>
          <xs:documentation>data Sets contain a heterogeneous collection of one or
more dataUnits</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="uid" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>uid of dataSet. </xs:documentation>
```



```
        </xs:annotation>
       </xs:element>
       <xs:element name="name" type="tml:BindType" minOccurs="0">
        <xs:annotation>
         <xs:documentation>name of dataSet</xs:documentation>
        </xs:annotation>
       </xs:element>
       <xs:element name="variableName" type="tml:BindType" minOccurs="0">
        <xs:annotation>
         <xs:documentation>Name of mathematical term used in the
transformation equations.  Index of component is the order in the sequence in the
LDS structure.</xs:documentation>
        </xs:annotation>
       </xs:element>
       <xs:element name="numObjInSet" type="tml:BindType" minOccurs="0">
        <xs:annotation>
         <xs:documentation>number of subordinate sets and/or arrays.  default
1</xs:documentation>
        </xs:annotation>
       </xs:element>
       <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
         <xs:documentation>A dataSet contains a collection of one or more
heterogeneous data units or data arrays.  dataArray are discouraged from being
used within a dataSet when defining the logical data structure.</xs:documentation>
        </xs:annotation>
        <xs:element ref="tml:dataUnit"/>
        <xs:element name="dataArray" type="tml:DataArrayType"/>
       </xs:choice>
      </xs:sequence>
      <xs:attributeGroup ref="tml:uid_uidRef"/>
     </xs:complexType>
    </xs:element>
    <xs:element name="dataArray" type="tml:DataArrayType">
     <xs:annotation>
      <xs:documentation>a dataArray contains a homogeneous collection of one or
more dataSets or dataArrays</xs:documentation>
     </xs:annotation>
    </xs:element>
   </xs:choice>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
 </xs:complexType>
 <xs:complexType name="IdentificationType">
  <xs:sequence>
   <xs:element name="uid" type="tml:BindType" minOccurs="0">
    <xs:annotation>
     <xs:documentation>uid of registry object</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="name" type="tml:BindType" minOccurs="0"/>
   <xs:element name="description" type="tml:BindType" minOccurs="0"/>
   <xs:element ref="tml:complexity" minOccurs="0"/>
   <xs:element name="characterization" minOccurs="0">
    <xs:annotation>
```

```
        <xs:documentation>Do the tml descriptions comply with the TML Compliance
Rules</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="characterizedBy" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                <xs:element name="organization" type="tml:BindType" minOccurs="0"/>
                <xs:element name="email" type="tml:BindType" minOccurs="0"/>
                <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
                <xs:element name="date" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="validatedBy" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                <xs:element name="organization" type="tml:BindType" minOccurs="0"/>
                <xs:element name="email" type="tml:BindType" minOccurs="0"/>
                <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
                <xs:element name="date" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="calibration" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Do the TML descriptions accurately reflect actual
performance specifications</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="calibratedBy" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="name" type="tml:BindType" minOccurs="0"/>
                <xs:element name="organization" type="tml:BindType" minOccurs="0"/>
                <xs:element name="email" type="tml:BindType" minOccurs="0"/>
                <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
                <xs:element name="date" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
                  </xs:annotation>
```



```
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="validatedBy" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="tml:BindType" minOccurs="0"/>
            <xs:element name="organization" type="tml:BindType" minOccurs="0"/>
            <xs:element name="email" type="tml:BindType" minOccurs="0"/>
            <xs:element name="phone" type="tml:BindType" minOccurs="0"/>
            <xs:element name="date" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>ISO8601 dateTime stamp</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="SpatialCoordType">
 <xs:sequence>
  <xs:element ref="tml:spaceCoordSystem" minOccurs="0"/>
  <xs:element name="spaceRefSystem" type="tml:BindType" minOccurs="0">
   <xs:annotation>
    <xs:documentation>which spatial reference system (i.e. spatial datum) are
spatial coordinates referenced (relative) to.   Allowed values: transducer,
earthCentered, earthLocal, subject. If ref system is transducer or subject then
the uid of the transducer or subject must be identified in the refObjUidRef
element.</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="refObjUidRef" type="tml:BindType" minOccurs="0">
   <xs:annotation>
    <xs:documentation>If the spaceRefSystem element is a transducer or a
Sunbect, then this element will identify the particular Transducer or Subject.
This is the UID reference of the object which position coordinates are referenced
(relative) to.</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="spaceCoords" minOccurs="0" maxOccurs="unbounded">
   <xs:annotation>
    <xs:documentation>TCF set of positional (translations and rotations)
coordinates for each shape,  space separated real numbers.  Order of coordinates
shall be from lowest frequency to highest frequency, same as lds. Default
locations and orientations are zero</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:sequence>
     <xs:element name="posVelAccel" type="tml:BindType" minOccurs="0">
      <xs:annotation>
```

```
            <xs:documentation>Allowed Values: pos, vel, accel,  Default is
pos.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="coordName" type="tml:BindType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Allowed Values: x, y, z, Alpha, beta, rho, latitude,
longitude, altitude, omega, phi, kappa,</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="coords" type="tml:ValueType" minOccurs="0"/>
      </xs:sequence>
      <xs:attributeGroup ref="tml:uid_uidRef"/>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
 </xs:complexType>
 <!--Global Elements-->
 <xs:element name="accuracy">
  <xs:annotation>
   <xs:documentation>accuracy is in terms of the data value before adjustment by
mult and offset.   if a characteristic frame (i.e. number of values) of values of
accuracy, then each value corresponds to the corresponding Characteristic Frame
position or interval</xs:documentation>
  </xs:annotation>
  <xs:complexType>
   <xs:sequence>
    <xs:element name="type" type="tml:BindType" minOccurs="0">
     <xs:annotation>
      <xs:documentation>Allowed values: relative, absolute, systematic, random.
default is absolute</xs:documentation>
     </xs:annotation>
    </xs:element>
    <xs:element name="errorDistribution" type="tml:BindType" minOccurs="0">
     <xs:annotation>
      <xs:documentation>Allowed Values: gaussian, chi, chi2, possion,  gamma.
default is gaussian</xs:documentation>
     </xs:annotation>
    </xs:element>
    <xs:element name="factor" type="tml:BindType" minOccurs="0">
     <xs:annotation>
      <xs:documentation>allowed values: 1sigma, 2sigma, 3sigma, 4sigma, 5sigma,
6sigma, percent, range. RMS, RSS, Default is 1sigma</xs:documentation>
     </xs:annotation>
    </xs:element>
    <xs:element name="accyValues" type="tml:BindType" minOccurs="0">
     <xs:annotation>
      <xs:documentation>A single accyValue relates to whole range of parent
coordinates (e.g. data or prop). If accyValue is variable over the parent
coordinates then there shall be a one-to-one correspondence between the accyValues
and the parent coordinates.  use mult and offset to describe variances over
CF</xs:documentation>
     </xs:annotation>
    </xs:element>
   </xs:sequence>
```



```
    </xs:complexType>
  </xs:element>
  <xs:element name="cfSubSampling">
    <xs:annotation>
      <xs:documentation>the CFSubSampling can be used for chipping part of a large
dataArry out and for reducing the number of points within an array for which to
associate modeling parameters.  there is one sub sampling element set of points
for each component structure (col, row, plane).    index numbers of col, row or
plane position within the CFs are listed for which corresponding modeling points
will be associated.  sample points are separated by commas, ranges are indicated
by ... between numbers which indicates a continuous interval for a single sample.
interpolation between samples uses logical structure.  </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="cfStructComp" type="tml:BindType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Allowed values: column, row, plane.  default is column.
One cfSubSampling element for each cfStructComp required.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="numOfSubSampleIndexPoints" type="tml:BindType"
minOccurs="0">
          <xs:annotation>
            <xs:documentation>Allowed values: positive integers from 1 to the number
of columns, rows, or planes in the data structure.  This number indicates the
number of samples in the cfSubSampleIndexPts.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="subSampleCfIndexPts" type="tml:BindType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>use same rules as points under value</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="tml:uid_uidRef"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="clusterDesc">
    <xs:annotation>
      <xs:documentation>An empty clusterdesc tag in a data stream indicates that
this cluster is no longer contained in the data stream.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element name="description" type="tml:BindType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>description of the data cluster</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="idMapping" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="tapPointUidRef" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
                <xs:annotation>
```

```
              <xs:documentation>dataUidRef of the tap point in the system to which
this cluster corresponds.  UID of the transducer, process input process output, or
connection node from which or to which this cluster relates.  This is the UID used
in the data header (i.e. reference attribute in data start tag).  Is some cases a
data in a single cluster may come from multiple dataUid tap
points.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="localID" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>short ID used in the data header (i.e. ref
attribute in data start tag)</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="clusterProperties" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="direction" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Allowed Values: fromSystem, toSystem.  default
fromSystem</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element ref="tml:complexity" minOccurs="0"/>
          <xs:element name="clusterType" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Allowed values: binary, packedXML.  verboseXML.
default binary</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="clusterSize" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Integer number of bytes in
Cluster</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="binHeaderEncode" minOccurs="0">
      <xs:annotation>
        <xs:documentation>If cluster type is binary this field describes the
encoding of the header attributes. binary files will contain only the contents of
the attributes and not the attribute tag.  The binary header will not contain the
left carrot and the letters "data" at the beginning of the header either, nor the
right carrot at the end of the header.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="headerAttrib" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>ref, reference, dateTime, contents and ismClass
attributes will be encoded and handled as "string" type</xs:documentation>
```

```
            </xs:annotation>
            <xs:complexType>
              <xs:sequence>
                <xs:element name="headerAttribName" type="tml:BindType"
minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>Allowed values: ref, clk, reference, dateTime,
contents, seq, total, ismClass. Default ref</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="dataType" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>Allowed values: text, number.  Default is
number. </xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="dataUnitFieldSize" minOccurs="0">
                  <xs:complexType>
                    <xs:choice>
                      <xs:sequence>
                        <xs:annotation>
                          <xs:documentation>if fixed field size.  Default is fixed
size.</xs:documentation>
                        </xs:annotation>
                        <xs:element name="numBits" type="tml:BindType" minOccurs="0">
                          <xs:annotation>
                            <xs:documentation>number of bits. default 8
</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                        <xs:element name="numSigBits" type="tml:BindType"
minOccurs="0">
                          <xs:annotation>
                            <xs:documentation>number of significant bits. default
8</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                        <xs:element name="justification" type="tml:BindType"
minOccurs="0">
                          <xs:annotation>
                            <xs:documentation>if numSigBits is less than numBits this
element indicates how sigbit are justified.  Allowed values: left, right. Default:
right</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                      </xs:sequence>
                      <xs:sequence>
                        <xs:annotation>
                          <xs:documentation>if field is variable size.
</xs:documentation>
                        </xs:annotation>
                        <xs:element name="beginTextDelimiter" type="tml:BindType"
minOccurs="0">
                          <xs:annotation>
```

```
                        <xs:documentation>delimiter used to separate variable size
dataUnits in cluster when encode is text (utf or ucs). default delimiter is none.
empty tag means none.</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="endTextDelimiter" type="tml:BindType"
minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>delimiter used to separate variable size
dataUnits in cluster when encode is text (utf or ucs). default delimiter is none.
Empty tag means none</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:choice>
              </xs:complexType>
            </xs:element>
            <xs:element name="endian" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Allowed values: big, little.  default
little</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="encode" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Allowed values: ucs16, utf8, signInt,
unsignInt, real,  bcd.  default unsignInt. </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="numBase" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>when numbers are encoded as text the number
base must be understood.  Allowed values: 2, 8, 10, 16, 32, 64, 128.  default
10</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="handleAsType" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>how should the text or number be handled in
the client application.  Allowed values: anuURI, boolean, byte, double, float,
short, string, int, integer, long, nonNegativeInteger, nonPositiveInteger,
positiveInteger,  unsignedByte, unsignedInt, unsignedShort,
unsignedLong.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
          <xs:attributeGroup ref="tml:uid_uidRef"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="timeTag" minOccurs="0">
  <xs:annotation>
```



```
        <xs:documentation>describes what time tag is used for the cluster.
Useful when parent systems normalize clocks from child components.  This element
also describes how accurately the sysClk value is applied to the cluster start
instant.  This is different from the accuracy of the system
clock.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="sysClkUidRef" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>if clk is used in the start tag and multiple clocks
are used in a system.  Default is the first parent system clock</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element ref="tml:accuracy" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="dataUnitEncoding" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>This unit describes the encoding of the dataUnit
identified in the dataUnitUidRef child element.  Some clusters which represent
only an event from a source or a trigger are empty and may not contain any
dataUnits.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="dataUnitUidRef" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>UID of the dataUnit from the logical structure.
</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="dataType" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Allowed values: text, number, binBlob.  Default is
text. </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="dataUnitFieldSize" minOccurs="0">
            <xs:complexType>
              <xs:choice>
                <xs:sequence>
                  <xs:annotation>
                    <xs:documentation>if fixed field size</xs:documentation>
                  </xs:annotation>
                  <xs:element name="numBits" type="tml:BindType" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>default  8</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="numSigBits" type="tml:BindType" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>default</xs:documentation>
                    </xs:annotation>
                  </xs:element>
```

```
                  <xs:element name="justification" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>if numSigBits is less than numBits this
element indicates how sigbit are justified.  Allowed values: left, right. Default:
right</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
                <xs:sequence>
                  <xs:annotation>
                    <xs:documentation>if field is variable size. Default is variable
size.</xs:documentation>
                  </xs:annotation>
                  <xs:element name="beginTextDelimiter" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>delimiter used to separate variable size
dataUnits in cluster when encode is text (utf or ucs). default delimiter is none.
Empty tag means none.</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="endTextDelimiter" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>default delimiter is none. Empty tag means
none.</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element name="endian" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Allowed values: big, little.  default
little</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="encode" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Allowed values: ucs16, utf8, signInt, unsignInt,
real, complex, bcd.  default utf8.  When clusterType is not binary only utf8 is
allowed in cluster.  All types are allowed when clusterType is binary. Complex
values are exchanged as two phenomenon (mag and phase or real and imaginary
components) or as a single complex number.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="numBase" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>when numbers are encoded as text the number base
must be understood.  Allowed values: 2, 8, 10, 16, 32, 64, 128.  default
10</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="handleAsType" type="tml:BindType" minOccurs="0">
```



```
            <xs:annotation>
              <xs:documentation>how should the text or number be handled in the
client application.  Allowed values: anuURI, boolean, byte, double, float, short,
string, int, integer, long, nonNegativeInteger, nonPositiveInteger,
positiveInteger,  unsignedByte, unsignedInt, unsignedShort,
unsignedLong.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:complexType>
    </xs:element>
    <xs:element name="numCfInCluster" type="tml:BindType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>number of characteristic frames in a cluster or the
number of clusters which comprise a large characteristic frame.  default = 1.
example: 2 means 2 CF per cluster, -2 means 2 clusters per CF.  Allowed values:
signed integer.  zero not allowed.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="transSeq" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>This is the order in which data is sent in the cluster
or CF (whichever is larger) relative to the logical data structure.  The order of
structure components are listed from lowest freq to highest frequency order.   If
transport sequence is blank then the sequence is the same as the logical order
(sequence) for that structure component.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="seqOfThisDataStruct" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>Sequence of (in this element - seqOfThisDataStruct)
in the data structure identified in the next element (inThisDataStruct).
seqOfBitsInUnit,  seqOfUnitsInSets, seqOfSetsInCf, seqOfCfInClust. Identify the
dataStructComponent in this element by dataUidRef.  dataUid of the cluster is
"cluster"</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="inThisDataStruct" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>Sequence of the data structure components
identified in the previous element (seqOfThisDataStruct) in the data structure
identified in this element (inThisDataStruct). seqOfBitsInUnit,  seqOfUnitsInSets,
seqOfSetsInCf, seqOfCfInClust. Identify the dataStructComponent in this element by
dataUidRef.  dataUid of the cluster is "cluster"</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="sequence" minOccurs="0">
            <xs:annotation>
```

```
              <xs:documentation>Allowed values; The sequence shall contain a string
of value separated by a comma.  Each value can be a positive integer or a range.
ranges shall be indicated by two integer numbers separated by three sequential
decimal points (....) to indicate a run from the first number to the
second</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="tml:BindType">
                  <xs:attributeGroup ref="tml:uid_uidRef"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
  <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
 </xs:complexType>
</xs:element>
<xs:element name="data">
 <xs:annotation>
  <xs:documentation>this element carries the date to or from transducer systems.
The data element will carry a single instance or a continuous stream of a
condition or set of synchronous conditions time tag to the precise instant of
creation.   There is no XML markup of data within the data tag.  A system
description will describe the decoding and understanding of the data within the
data tag.</xs:documentation>
 </xs:annotation>
 <xs:complexType>
  <xs:simpleContent>
   <xs:extension base="xs:string">
    <xs:attribute name="ref" type="xs:string" use="optional">
     <xs:annotation>
      <xs:documentation>alias or short id reference of transducer or process
producing this data</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="clk" type="xs:integer" use="optional">
     <xs:annotation>
      <xs:documentation>sys clock state at trigger point to data cluster.  For
low sampling frequency transducers this high frequency clock state may not be
required.  A full dateTime attribute may suffice for time synchronization of
data.</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="reference" type="xs:anyURI" use="optional">
     <xs:annotation>
      <xs:documentation>this is the full UID reference to the cluster
description</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="dateTime" type="xs:dateTime" use="optional">
     <xs:annotation>
```



```
          <xs:documentation>Full qualified date and time of transducer or process
producing this data. For low sampling frequency transducers this high frequency
clock state may not be required.  A full dateTime attribute may suffice for time
synchronization of data.</xs:documentation>
         </xs:annotation>
        </xs:attribute>
        <xs:attribute name="contents" use="optional">
         <xs:annotation>
          <xs:documentation>If a binary stream header does not contain a contents
field then the data cluster is by default explicit data.  This field is encoded as
a binary (2-bits) "00" in a binary file if the field is
contained.</xs:documentation>
         </xs:annotation>
         <xs:simpleType>
          <xs:restriction base="xs:string">
           <xs:enumeration value="exp"/>
           <xs:enumeration value="imp"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="seq" type="xs:integer" use="optional">
         <xs:annotation>
          <xs:documentation>if no "total" attribute exist then this attribute can
be used to number the data elements like a count, this enables the receipt end to
determine if any data clusters were lost.</xs:documentation>
         </xs:annotation>
        </xs:attribute>
        <xs:attribute name="total" type="xs:integer" use="optional">
         <xs:annotation>
          <xs:documentation>total in sequence e.g. 1 of 4, 2 of 4.  1 and 2 being
the seq number and 4 being the total</xs:documentation>
         </xs:annotation>
        </xs:attribute>
        <xs:attribute name="ismclass" type="ism:ClassificationType">
         <xs:annotation>
          <xs:documentation>security classification of each data cluster. Overall
data classification of  transducer data in clusterDescription.  Overall
classification of file or stream in tml start tag.</xs:documentation>
         </xs:annotation>
        </xs:attribute>
       </xs:extension>
      </xs:simpleContent>
     </xs:complexType>
    </xs:element>
    <xs:element name="dataUnit">
     <xs:annotation>
      <xs:documentation>an elemental unit of data.  one description for each
unit</xs:documentation>
     </xs:annotation>
     <xs:complexType>
      <xs:sequence>
       <xs:element name="uid" type="tml:BindType" minOccurs="0">
        <xs:annotation>
         <xs:documentation>uid of dataUnit</xs:documentation>
        </xs:annotation>
       </xs:element>
```

```
      <xs:element name="name" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>name of dataUnit</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="variableName" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Name of mathematical term used in the transformation
equations.  Index of component is the order in the sequence in the LDS
structure.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="dataType" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Allowed values: number, complexNumber, text, or
binaryBlob.  default is number</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="bytesInBlob" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>If dataType is binaryBlob then number of bytes in the
binary blob.  Not used for transducer structures, only for process
structures.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="tml:uid_uidRef"/>
  </xs:complexType>
</xs:element>
<xs:element name="logicalDataStructure">
  <xs:annotation>
    <xs:documentation>the logical structure of data (i.e. of the characteristic
frame).  This is not necessarily the structure or order that data is communicated
in.  The transmission order is defined in the cluster description.  The
transmission order is defined relative to the logical order.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="uid" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>uid of lds</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="name" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>name of lds</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ldsDimensionality" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Allowed values: 0, 1, 2, 3.  Default is 0.
dimensionality of the logical data structure (lds).  number of structure
components used for giving hints for data representation.  0 dim is a single
value, 1 dim is a series of columns, rows or planes, 2 dim is any order of  two
structure components (col-row, col-plane, or row-plane), and a 3 dim is any order
of three structure components col-row-plane</xs:documentation>
```



```
        </xs:annotation>
      </xs:element>
      <xs:element name="numOfDataSetsInCf" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Number of dataSets or dataArrays in the Characteristic
Frame.  Allowed Value: positive integer.  Default:1</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="cfDataArray" type="tml:DataArrayType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>logical data structure of the characteristic frame.
Lowest frequency array first.</xs:documentation>
        </xs:annotation>
      </xs:element>
     </xs:sequence>
     <xs:attributeGroup ref="tml:uid_uidRef"/>
   </xs:complexType>
  </xs:element>
  <xs:element name="objToDataRelation">
   <xs:annotation>
     <xs:documentation>Connects transducer to bindUids.  Associate transducer data
to a (remote) object.  This may occur after data acquisition. An object is either
a transducer, subject or their properties.    Many subjects may be related to data
in a dataArray. The objects can be related to data units, sets and arrays to
subjects.  </xs:documentation>
   </xs:annotation>
   <xs:complexType>
     <xs:sequence>
      <xs:element name="relationDescription" minOccurs="0">
        <xs:annotation>
          <xs:documentation>description of the signal or the property
relation</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="tml:BindType"/>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="uid" type="tml:BindType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>connection or node UID of the connection signal data or
property relationship</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="object" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Object can be a single transducer (dangle relation), a
single dataUID, or many subjects can be related to a single data unit.
probabilities can be assigned to each relation.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>name of the object</xs:documentation>
```

RETIRED

```
            </xs:annotation>
          </xs:element>
          <xs:element name="objType" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>identify object as a transducer or a subject.
Allowed Values: subject, transducer. Default: subject</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element ref="tml:cfSubSampling" minOccurs="0"/>
          <xs:element name="objUidRef" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>UID of the object (subject or transducer, or
probable subject).  local id of the subject if multiple ids are used to associate
with each cell of  the logical structure.  </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="objLocalID" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>if localId assigned to objUidRef for building CF of
obj to data (i.e.CF) relationships. Sequence of values is the same as the sequence
in the data (logical data structure or subsampled data structure, if
present)</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="confidence" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Value range -1 to 1.  -1 is 100% no confidence.
confidence values match same sequence as logical data structure or subsampled data
structure, if present (if multiple objects in data structure)</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:complexType>
    </xs:element>
    <xs:choice>
      <xs:sequence>
        <xs:element name="name" type="tml:BindType" minOccurs="0"/>
        <xs:element name="dataUidRef" type="tml:BindType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>UID of the data reference.  Archived data streams
will have a UID indicative of the data source, time, and clk count of the start.
</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
      <xs:element name="value" type="tml:BindType" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
  <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
 </xs:complexType>
</xs:element>
<xs:element name="objToObjRelation">
 <xs:annotation>
```



```
    <xs:documentation>This relation describes object to object relations.
Attaching a transducer to an object (object is a subject or a transducer) (i.e.
dangle, where the only thing the transducer interfaces to is that subject. (cant
different individual data many measures with many individual subjects, see
objToData). The transducer to transducers relation does not include phenomenon to
phenomenon connections, see dataToData</xs:documentation>
   </xs:annotation>
   <xs:complexType>
    <xs:sequence>
     <xs:element name="relationDescription" type="tml:BindType" minOccurs="0">
      <xs:annotation>
       <xs:documentation>description of the  relation</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="uid" type="tml:BindType" minOccurs="0">
      <xs:annotation>
       <xs:documentation>uid of the relationship</xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element name="object" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
       <xs:documentation>many objects can be related to a many objects.
probabilities can be assigned to each relation</xs:documentation>
      </xs:annotation>
      <xs:complexType>
       <xs:sequence>
        <xs:element name="name" type="xs:string" minOccurs="0">
         <xs:annotation>
          <xs:documentation>name of the object</xs:documentation>
         </xs:annotation>
        </xs:element>
        <xs:element name="objType" type="tml:BindType" minOccurs="0">
         <xs:annotation>
          <xs:documentation>identify object as a transducer or a subject.
Allowed Values: subject, transducer. Default: subject</xs:documentation>
         </xs:annotation>
        </xs:element>
        <xs:element name="dirIndirSubj" type="tml:BindType" minOccurs="0">
         <xs:annotation>
          <xs:documentation>if objType is subject then identify if direct or
indirect subject.  Allowed values: direct, indirect.  Default is
direct.</xs:documentation>
         </xs:annotation>
        </xs:element>
        <xs:element name="objUidRef" type="tml:BindType" minOccurs="0">
         <xs:annotation>
          <xs:documentation>UID of the subject (or probable subject).  local id
of the subject if multiple ids are used to associate with each cell of  the
logical structure.  Sequence of values is the same as the sequence in the data
(logical data structure)</xs:documentation>
         </xs:annotation>
        </xs:element>
       </xs:sequence>
       <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:complexType>
     </xs:element>
```

```
        <xs:element name="confidence" type="tml:BindType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>confidence of relationship (-1 to 1). -1 is 100% no
confidence. confidence values match same sequence as logical data structure (if
multiple values in data structure)</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="ism:SecurityAttributesOptionGroup"/>
      <xs:attributeGroup ref="tml:uid_uidRef"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="process" type="tml:ProcessType">
    <xs:annotation>
      <xs:documentation>A transducer can be a stand alone object or part of a
system. Describes derivation of output dataUnits relative to input dataUnits or
constants.  An empty process tag in a data stream indicates that this process is
no longer a part of the system</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="responseModels">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="tml:cfSubSampling" minOccurs="0"/>
        <xs:element name="steadyStateResponse" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>input to output  mapping.  one or more mappings for
each dataUnit.  Can have property-property, property-data, or data-property
mappings.  property-property-property and property-property-data mappings are also
allowed as long as independent property values can be found somewhere.  Separate
mappings can be used for different hystersis directions or for non-continuous or
broken functions.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="responseParameters" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="codePlot" type="tml:BindType" minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>Allowed values code, plot. Default:
plot</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="hysteresisDirection" type="tml:BindType"
minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>allowed values: increasing, decreasing, both.
default both</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="calibrated" type="tml:BindType" minOccurs="0">
                      <xs:annotation>
                        <xs:documentation>Is response calibrated, or is response a
relative reading? true of false. Default: true</xs:documentation>
                      </xs:annotation>
```



```
                </xs:element>
                <xs:element name="proportional" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>For uncalibrated responses is the output
proportional to the input? true of false. Mult factors can also reflect prop or
inversely prop for calibrated responses. Default: true.</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="invertability" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>a process input can be determined from its
output. Allowed Values: true, false.  default true</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="timeInvariant" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>a time shift in the input only results in a
time shift in the output. Allowed Values: true, false.  default
true</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="linear" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>allowed values: true or false.  do not need
explicit Phen plot values if linear is true. Phen and data mult and offset can be
used if there are no limits.  default true</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="interCfInterpolate" type="tml:BindType"
minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>Allowed values: continuous, discrete,
lastValue, returnToZero.  how to interpolate between corresponding data values
between adjacent CF's.  default is continuous</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="intraCfInterpolate" type="tml:BindType"
minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>Allowed values: continuous, discrete,
lastValue, returnToZero.  how to interpolate between data values within a CF.
default continuous</xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="propValues" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>values for the physical property (phenomenon) axis
of the input output transfer function</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="tml:ValueType">
                  <xs:sequence>
```

```
                  <xs:element name="inputOutput" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Is the physical property (phenomenon) the
input or output for this dataUnit.  Allowed values: input, output.  Default:
input</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="propName" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>from Physical Property (Phenomenon)
Dictionary</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="tml:BindType">
                          <xs:attributeGroup ref="tml:uid_uidRef"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="propQualifier" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Qualifier for the property.  From Qualifier
Dictionary.  e.g. aveValue, rmsValue, rssValue, instValue, accumulatedValue,
rateOfChange, range, min, max...</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="UOM" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>From Units Of Measure Dictionary (SI
Units)</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:simpleContent>
                        <xs:extension base="tml:BindType">
                          <xs:attributeGroup ref="tml:uid_uidRef"/>
                        </xs:extension>
                      </xs:simpleContent>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="direction" type="tml:BindType" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>if the physical property (phenomenon) had a
direction associated with it such as torque or force. direction relative to the
transducer reference system.  Allowed Values: horizontal, vertical, +xTranslation,
-xTranslation, +yTranslation, -yTranslation, +zTranslation, -zTranslation, +alpha,
-alpha, +beta, -beta, +rhoTranslation, -rhoTranslation, +latTranslation, -
latTranslation, +longTranslation
longTranslation, +altTranslation, -altTranslation, +omegaRotation, -omegaRotation,
+phiRotation, -phiRotation, +kappaRotation, -kappaRotation, none Default:
none</xs:documentation>
                    </xs:annotation>
                  </xs:element>
```



```
                  <xs:element name="variableName" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Name of mathematical term used in the
transformation equations.  </xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="calibProp" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
                    <xs:annotation>
                      <xs:documentation>If a calibrated source is available this
elements identifies the calibration level or points (bindUID) to the calibrated
sensor measuring the source.  This is used for post correcting relative readings
Default: none</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="dataValues" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>values for the data axis of the input output
transfer function</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="tml:ValueType">
                <xs:sequence>
                  <xs:element name="inputOutput" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Is the data an input or an output for this
dataUnit.  Allowed values: input, output.  Default: output</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="dataUidRef" type="tml:BindType" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>uid of the data form the logical data
structure (dataUnit) to which this response model corresponds</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="variableName" type="tml:BindType"
minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Name of mathematical term used in the
transformation equations.  </xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="calibData" type="tml:BindType" minOccurs="0"
maxOccurs="unbounded">
                    <xs:annotation>
                      <xs:documentation>data resulting from calibrated source. or
bindUID points to sensor measurement measuring calib source. Default:
none</xs:documentation>
```

```
                </xs:annotation>
              </xs:element>
            </xs:sequence>
          </xs:extension>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="code" minOccurs="0">
      <xs:annotation>
        <xs:documentation>computer code of the transfer process from input to
output</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="properties" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="codeType" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>Allowed Values: source, exe default:
source</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element name="codeLanguage" type="tml:BindType"
minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>Allowed Values: C, C++, Java, Fortran, C
Sharp, Basic, Visual Basic. Default: C</xs:documentation>
                  </xs:annotation>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="listing" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Listing of code. Base64 encoded executable or
source code with unallowed XML characters escaped out</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:simpleContent>
                <xs:extension base="tml:BindType"/>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
</xs:complexType>
</xs:element>
<xs:element name="impulseResponse" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
```



```
          <xs:documentation>time domain or frequency domain impulse characteristics
for linear time invariant transforms. May have a separate response for each
dataUnit and for each type (freq and time).  Or dataUnits within a data Set may
share the same response.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="dataUidRef" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>same as uidRef in attributes</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="freqTime" type="tml:BindType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Allowed values: freq, time.  default is time.
indicates if frequency of time domain descriptions.  </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="amplitude" minOccurs="0">
              <xs:annotation>
                <xs:documentation>amplitude dependent axis.</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="tml:ValueType"/>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="time" minOccurs="0">
              <xs:annotation>
                <xs:documentation>time domain independent axis.</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:complexContent>
                  <xs:extension base="tml:ValueType"/>
                </xs:complexContent>
              </xs:complexType>
            </xs:element>
            <xs:element name="frequency" type="tml:ValueType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>frequency domain independent
axis.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
          <xs:attributeGroup ref="tml:uid_uidRef"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="frequencyResponse" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>one for each dataUnit and for each type of freqResp
(carrier, modulation, and powerSpectrialDensity) and each type of plot amp vs freq
and phase vs freq (can combine plots onto one as well)</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
```

```
          <xs:element name="dataUidRef" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>same as uidRef in attributes</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="freqRespType" type="tml:BindType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Allowed values: carried, modulation, PSD
(pwrSpectralDensity).  default carrier</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="amplitude" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Set of point coordinates describing amplitude
dependent axis</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="tml:ValueType"/>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="phase" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Set of point coordinates describing phase dependent
axis</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="tml:ValueType"/>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
          <xs:element name="frequency" minOccurs="0">
            <xs:annotation>
              <xs:documentation>Set of point coordinates describing frequency
independent axis</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="tml:ValueType"/>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
</xs:complexType>
</xs:element>
<xs:element name="spaceCoordSystem" type="tml:BindType">
  <xs:annotation>
    <xs:documentation>Allowed values: spherical,  rectangular, cylindrical,
wgs84elliptical.  default is spherical.</xs:documentation>
```



```
   </xs:annotation>
  </xs:element>
  <xs:element name="spatialModel">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="dataUidRef" type="tml:BindType" minOccurs="0">
      <xs:annotation>
       <xs:documentation>corresponding UID of dataUnit, dataSet,  or data Array.
If data array then all subordinate data structures share same model (row, col, or
plane), if dataSet then all data units share same model (cf), if dataUnit then
only that units model is described (cf). </xs:documentation>
      </xs:annotation>
     </xs:element>
     <xs:element ref="tml:cfSubSampling" minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="ambiguitySpace" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
       <xs:documentation>Multiple AS are combined as spatial intersections.
e.g. one for columns and one for rows.  Typically every cell within a multiple
cell CF will share the same shape but have unique positions.</xs:documentation>
      </xs:annotation>
      <xs:complexType>
       <xs:sequence>
        <xs:element name="shape" minOccurs="0" maxOccurs="unbounded">
         <xs:annotation>
          <xs:documentation>This is the shape of the AS for the power profile
indicated.  May also have multiple shapes to define multiple lobes of energy
fields.  Multiple shapes within an AS are combined as a spatial unions.   The
position elements defines the position of each shape.</xs:documentation>
         </xs:annotation>
         <xs:complexType>
          <xs:sequence>
           <xs:element name="pwrProfile" minOccurs="0">
            <xs:annotation>
             <xs:documentation>The equi-power surface power level compared to
the point of transmission or reception.   default is -3db beam pattern,
pwrProfile="-3".</xs:documentation>
            </xs:annotation>
            <xs:complexType>
             <xs:simpleContent>
              <xs:extension base="tml:BindType"/>
             </xs:simpleContent>
            </xs:complexType>
           </xs:element>
           <xs:element ref="tml:spaceCoordSystem" minOccurs="0"/>
           <xs:element name="spaceLocCoords" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
             <xs:documentation>one set of coordinates for each spatial axes.
Each shape is defined relative to an arbitrary data spatial reference system.
</xs:documentation>
            </xs:annotation>
            <xs:complexType>
             <xs:sequence>
              <xs:element name="coordName" type="tml:BindType" minOccurs="0">
               <xs:annotation>
```

```
                          <xs:documentation>Allowed values: x, y, z, alpha, beta, rho.
</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="coords" type="tml:ValueType" minOccurs="0">
                        <xs:annotation>
                          <xs:documentation>values contains a string of real numbers.
The mult and offset are single values, unless the shape varies over the
Characteristic Frame then the mult and offset may contain a Characteristic Frame
array of values. simple IFOV alpha=0, beta=0.  (ray where rho is
infinite)</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                    </xs:sequence>
                    <xs:attributeGroup ref="tml:uid_uidRef"/>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
              <xs:attributeGroup ref="tml:uid_uidRef"/>
            </xs:complexType>
          </xs:element>
          <xs:element name="position" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>location and attitude of ambiguity
shape</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:complexContent>
                <xs:extension base="tml:SpatialCoordType"/>
              </xs:complexContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="tml:uid_uidRef"/>
</xs:complexType>
</xs:element>
<xs:element name="subject">
  <xs:annotation>
    <xs:documentation>This is the subject (object, thing) that relates to the
phenomenon (property) that is affected or detected by the transducer. The relation
between a subject and transducer data or subject and subject is described in the
relationship element. An empty subject tag in a data stream indicates that this
object is no longer a part of the system</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="xs:anyType">
        <xs:attributeGroup ref="tml:uid_uidRef"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```



```
  <xs:element name="system" type="tml:SystemType">
    <xs:annotation>
      <xs:documentation>An empty system tag (with id) in a data stream indicates
that the system is no longer available in the stream, or if system was not
previously part of the parent system it will be added to the parent
system.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="temporalModel">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="cfTrigger" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="trigType" type="tml:BindType" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>Allowed Values: private, privateOnDataRecipt,
privateOnInputTrig, pvtOnChgOutput.   publicOnTrigReciept.   public trigger:
controllable by external commands. private trigger: uncontrollable by external
commands.  Virtual trig sensor puts sysClk time in data tag.  If public a bindUid
is made available.  default trigger is privatePeriodic.</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="publicTrigger" type="tml:BindType" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>if trigger is public then this identifies the
uidRef of trigger source (command).  Whenever a data cluster is sent to this UID
or to the uid of a process that is bound to this uid then this transducer or
process cycle will trigger.  The bindUid enables late binding of the trigger
source</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="period" type="tml:ValueType" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>if private trigger is periodic then,  trigger
period in seconds</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:sequence>
            <xs:attributeGroup ref="tml:uid_uidRef"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="cfDuration" type="tml:ValueType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>time duration of the CF in seconds.  Can also be
determined by the CF offset time values by subtracting the smallest offset time
from the largest offset time.  Duration does not vary over the CF.  Only one
value.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="latencyTime" type="tml:ValueType" minOccurs="0">
          <xs:annotation>
```

```
          <xs:documentation>latency time in seconds (real number).  Time between
the input and the output.  Transducer time tags should be corrected to reflect
correct input time for receivers and output time for transmitters.  Latency for
processes reflects the process delay.  Latency time does not vary over the CF.
Only one value. </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ambiguityTime" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>data integration time for each sample in the CF. Each
dataunit may have a different time.   This element contains the number of samples
in a CF or the number indicated by the noOfSubSampledIndexPoints element in the
CFsubSamplingSequence or just one time.  If just one time then the same time
applies to all sample in the CF.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tml:ValueType">
              <xs:sequence>
                <xs:element name="dataUidRef" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>corresponding UID of dataUnit or dataSet.
Duplicate of uid in identification element Default: Uid of
dataSet</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element ref="tml:cfSubSampling" minOccurs="0"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="cfOffsetTime" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>cfOffSetTime contains time offsets for each dataUnit or
dataSet in the CF relative to the clock attribute (clk or dateTime) in the data
start tag.  contains the number of time values indicated by the
numSubSampledIndexPoints in the cfSubSampling child element. or
num</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="tml:ValueType">
              <xs:sequence>
                <xs:element name="dataUidRef" type="tml:BindType" minOccurs="0">
                  <xs:annotation>
                    <xs:documentation>corresponding UID of dataUnit or dataSet.
Duplicate of uid in identification element Default: Uid of
dataSet</xs:documentation>
                  </xs:annotation>
                </xs:element>
                <xs:element ref="tml:cfSubSampling" minOccurs="0"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:extension>
```



```
      </xs:complexContent>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
   <xs:attributeGroup ref="tml:uid_uidRef"/>
  </xs:complexType>
 </xs:element>
 <xs:element name="transducer" type="tml:TransducerType">
  <xs:annotation>
   <xs:documentation>A transducer can be a stand alone object or part of a
system.  An empty transducer tag in a data stream indicates that this transducer
is no longer a part of the system</xs:documentation>
  </xs:annotation>
 </xs:element>
 <xs:attributeGroup name="uid_uidRef">
  <xs:attribute name="name" type="xs:string" use="optional">
   <xs:annotation>
    <xs:documentation>short descriptive name of element</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="uid" type="xs:anyURI" use="optional">
   <xs:annotation>
    <xs:documentation>unique ID for this element</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:attribute name="uidRef" type="xs:anyURI" use="optional">
   <xs:annotation>
    <xs:documentation>the contents of this element are exactly the same as the
contents of the uidRef element. no need repeating it. (similar to
xlink)</xs:documentation>
   </xs:annotation>
  </xs:attribute>
 </xs:attributeGroup>
 <xs:element name="complexity" type="tml:BindType">
  <xs:annotation>
   <xs:documentation>indication of the complexity of handling this data. Allowed
Values: 1A - 1F, 2A -2F, 3A - 3F, 4A - 4F, 5A - 5F.  default 1A</xs:documentation>
  </xs:annotation>
 </xs:element>
</xs:schema>
```

Annex B
(Normative)

Abstract test suite

B.1 General

A validation and verification service will confirm the proper implementation of TML messages. A description of the test suite will be described in latter versions of TML.

Annex C
(Informative)

Example TML Instance Documents

Example TML instanced documents will be posted on the following site

http://www.ogcnetwork.net/node/105

Bibliography

Manual of Photogrammetry, American Society of Photogrammetry, Fifth Edition, 2004

Optical & Photographic Reconnaissance Systems, Niels Jensen, 1968

Handbook of Modern Sensors – Physics, Designs, and Applications (3rd Edition), Jacob Fraden, 2003

Handbook of Measurement Science, Volume 1 Theoretical Fundamentals, Peter Sydenham, 1992

Handbook of Measurement Science, Volume 2 Practical Fundamentals, Peter Sydenham, 1992

Handbook of Measurement Science, Volume 3 Elements of Change, Peter Sydenham, 1992

Acoustical Imaging Volume 19, Proceedings of the 19th International Symposium on Acoustical Holography, 1991

Fundamentals of Electronic Instrumentation for Measurement, First Edition, William Ribbins, 1973

Introduction to Airborne Radar, George Stimson, 1983

Multisensor Data Fusion, Ed Waltz, James Llinas, 1990

Mathematical Techniques in Multisensor Data Fusion, David Hall, 1992

Elements of Photogrammerty, Paul Wolf, 1974

Introduction to Radar Systems, Merrill Skolink, 1980

Synthetic Aperture Radar Signal Processing, Mehrdad Soumekh, 1999

RADAR Handbook (Second Edition), Merrill Skolnik, 1990

Radargrammetric Image Processing, Franz Leberl, 1989

Communication and Radar Systems, Nicolaos Tzanes, 1985

Synthetic Aperture Radar – Systems and Signal Processing, John Curlander, Robert McDonough, 1991

Signals, Systems and Transforms, Charles Phillips, John Parr, 1995

Introduction to Strapdown International Navigation Systems, Paul Savage, 1985

DMA TR 8350.2-B Supplement to DOD WGS-84 Technical Report – Parameters, Formulas, and Graphics for the Practical Application of WGS-84, 1987

Proposed Replacement Sensor Model (RSM) NITF TRE, Draft 7/23/2004

