MTConnect® Standard

Part 2.0 – Device Information Model

Version 2.5.0

Prepared for: MTConnect Institute
Prepared from: `MTConnectSysMLModel.xml`
Prepared on: January 5, 2026

MTConnect® is a registered trademark of AMT - The Association for Manufacturing Technology. Use of MTConnect is limited to use as specified on `http://www.mtconnect.org/`.

January 5, 2026

MTConnect Specification and Materials

The Association for Manufacturing Technology (AMT) owns the copyright in this MTConnect Specification or Material. AMT grants to you a non-exclusive, non-transferable, revocable, non-sublicensable, fully-paid-up copyright license to reproduce, copy and redistribute this MTConnect Specification or Material, provided that you may only copy or redistribute the MTConnect Specification or Material in the form in which you received it, without modifications, and with all copyright notices and other notices and disclaimers contained in the MTConnect Specification or Material.

If you intend to adopt or implement an MTConnect Specification or Material in a product, whether hardware, software or firmware, which complies with an MTConnect Specification, you shall agree to the MTConnect Specification Implementer License Agreement ("Implementer License") or to the MTConnect Intellectual Property Policy and Agreement ("IP Policy"'). The Implementer License and IP Policy each sets forth the license terms and other terms of use for MTConnect Implementers to adopt or implement the MTConnect Specifications, including certain license rights covering necessary patent claims for that purpose. These materials can be found at `www.MTConnect.org`, or or by contacting `mailto:info@MTConnect.org`.

MTConnect Institute and AMT have no responsibility to identify patents, patent claims or patent applications which may relate to or be required to implement a Specification, or to determine the legal validity or scope of any such patent claims brought to their attention. Each MTConnect Implementer is responsible for securing its own licenses or rights to any patent or other intellectual property rights that may be necessary for such use, and neither AMT nor MTConnect Institute have any obligation to secure any such rights.

This Material and all MTConnect Specifications and Materials are provided "as is" and MTConnect Institute and AMT, and each of their respective members, officers, affiliates, sponsors and agents, make no representation or warranty of any kind relating to these materials or to any implementation of the MTConnect Specifications or Materials in any product, including, without limitation, any expressed or implied warranty of noninfringement, merchantability, or fitness for particular purpose, or of the accuracy, reliability, or completeness of information contained herein. In no event shall MTConnect Institute or AMT be liable to any user or implementer of MTConnect Specifications or Materials for the cost of procuring substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, indirect, special or punitive damages or other direct damages, whether under contract, tort, warranty or otherwise, arising in any way out of access, use or inability to use the MTConnect Specification or other MTConnect Materials, whether or not they had advance notice of the possibility of such damage.

The normative XMI is located at the following URL: `MTConnectSysMLModel.xml`

Table of Contents

Table of Figures

List of Tables

1 Purpose of This Document

This document, *MTConnect Standard: Part 2.0 - Device Information Model* of the MTConnect Standard, establishes the rules and terminology to be used by designers to describe the function and operation of a piece of equipment and to define the data that is provided by an *agent* from the equipment. The *Device Information Model* also defines the structure for the *response document* that is returned from an *agent* in response to a *probe request*.

In the MTConnect Standard, equipment represents any tangible property that is used in the operations of a manufacturing facility. Examples of equipment are machine tools, ovens, sensor units, workstations, software applications, and bar feeders.

> Note: See *MTConnect Standard: Part 3.0 - Observation Information Model* of the MTConnect Standard for details on the *response document* that are returned from an *agent* in response to a *sample request* or *current request*.

2 Terminology and Conventions

Refer to *MTConnect Standard Part 1.0 - Fundamentals* for a dictionary of terms, reserved language, and document conventions used in the MTConnect Standard.

2.1 General Terms

adapter

optional piece of hardware or software that transforms information provided by a piece of equipment into a form that can be received by an *agent*.

agent

software that collects data published from one or more piece(s) of equipment, organizes that data in a structured manner, and responds to requests for data from client software systems by providing a structured response in the form of a *response document* that is constructed using the *semantic data model* of a Standard.

alarm limit

limit used to trigger warning or alarm indicators.

application

software or a program that is specific to the solution of an application problem. *Ref ISO/IEC 20944-1:2013*

archetype

archetype provides the requirements, constraints, and common properties for a type of *Asset*.

asset buffer

buffer for *Assets*.

attachment

connection by which one thing is associated with another.

buffer

section of an *agent* that provides storage for information published from pieces of equipment.

cartesian coordinate system

3D orthogonal coordinate system [(]ISO/IEC 19794-5:2011en).

characteristic

control placed on an element of a *feature* such as its size, location, or form, which may be a specification limit, a nominal with tolerance, or some other numerical or non-numerical control. *Ref QIF 3.0 3.4.29. Ref AS9102-B*.

client

application that sends *request* for information to an *agent*.

> Note: Examples include software applications or a function that implements the *request* portion of an *interface interaction model*.

combined standard uncertainty

standard uncertainty of the result of a measurement when that result is obtained from the values of a number of other quantities, equal to the positive square root of a sum of terms, the terms being the variances or covariances of these other quantities weighted according to how the measurement result varies with changes in these quantities. *Ref JCGM 100:2008 2.3.4*

condition activation

state transition from `Normal` to either `Warning` or `Fault`.

controlled vocabulary

restricted set of values for a given property.

data dictionary

listing of standardized terms and definitions used in *MTConnect Information Model*.

data model

organizes elements of data and standardizes how they relate to one another and to the properties of real-world entities.

data set

key-value pairs where each entry is uniquely identified by the *key*.

data source

piece of equipment that can produce data that is published to an *agent*.

deprecated

indication that specific content in an *MTConnect Document* is currently usable but is regarded as being obsolete or superseded.

deprecation warning

indication that specific content in an *MTConnect Document* may be changed to *deprecated* in a future release of the standard.

document

piece of written, printed, or electronic matter that provides information or evidence that serves as an official record.

electric current

rate of flow of electric charge.

element

constituent part or a basic unit of identifiable and definable data.

extensible

ability for an implementer to extend *MTConnect Information Model* by adding content not currently addressed in the MTConnect Standard.

feature

topological entity(ies) or design requirements related to a geometric model. *Ref QIF 3.0-3.4.59*

force

push or pull on a mass which results in an acceleration.

heartbeat

function that indicates to a *client* that the communications connection to an *agent* is still viable during times when there is no new data available to report often referred to as a "keep alive" message.

higher level

nested element that is above a lower level element.

implementation

specific instantiation of the MTConnect Standard.

information model

rules, relationships, and terminology that are used to define how information is structured.

instance

describes a set of *streaming data* in an *agent*. Each time an *agent* is restarted with an empty *buffer*, data placed in the *buffer* represents a new *instance* of the *agent*.

interaction model

model that defines how information is exchanged across an *interface* to enable interactions between independent systems.

interface

means by which communication is achieved between independent systems.

key

unique identifier in a *key-value pair* association.

key-value pair

association between an identifier referred to as the *key* and a value which taken together create a *key-value pair*.

location

place or named space associated with an object or that can be occupied by an object.

lower camel case

first word is lowercase and the remaining words are capitalized and all spaces between words are removed.

lower level

nested element that is below a higher level element.

lower limit

lower conformance boundary for a variable.

lower warning

lower boundary indicating increased concern and supervision may be required.

major

identifier representing a consistent set of functionalities defined by the MTConnect Standard.

maximum

numeric upper constraint.

message

communication in writing, in speech, or by signals.

metadata

data that provides information about other data.

minimum

numeric lower constraint.

minor

identifier representing a specific set of functionalities defined by the MTConnect Standard.

nominal

ideal or desired value for a variable.

organize

act of containing and owning one or more elements.

organizer

entity that *organizes* one or more elements.

parameter

variable that must be given a value during the execution of a program or a communications command.

part

discrete item that has both defined and measurable physical characteristics including mass, material, and features, and is created by applying one or more manufacturing process steps to a workpiece

pascal case

first letter of each word is capitalized and the remaining letters are in lowercase. All space is removed between letters

patch

supplemental identifier representing only organizational or editorial changes to a *minor* version document with no changes in the functionality described in that document.

persistence

method for retaining or restoring information.

position

location that is represented by a point in space relative to a reference.

probe

instrument commonly used for measuring the physical geometrical characteristics of an object.

profile

extends a reference metamodel (such as Unified Modeling Language (UML)) by allowing to adapt or customize the metamodel with constructs that are specific to a particular domain, platform, or a software development method.

requester

entity that initiates a *request* for information in a communications exchange.

reset

act of reverting back the accumulated value or statistic to their initial value.

Note: An *Observation* with a *data set* representation removes all *key-value pairs*, setting the *data set* to an empty set.

responder

entity that responds to a *request* for information in a communications exchange.

response document

electronic *document* published by an *MTConnect Agent* in response to a *probe request*, *current request*, *sample request* or *asset request*.

schema

definition of the structure, rules, and vocabularies used to define the information published in an electronic document.

semantic data model

methodology for defining the structure and meaning for data in a specific logical way that can be interpreted by a software system.

sensing element

mechanism that provides a signal or measured value.

sequence number

primary key identifier used to manage and locate a specific piece of *streaming data* in an *agent*.

specification limit

limit defining a range of values designating acceptable performance for a variable.

spindle

mechanism that provides rotational capabilities to a piece of equipment.

Note: Typically used for either work holding, materials or cutting tools.

standard

document established by consensus that provides rules, guidelines, or characteristics for activities or their results.. *Ref ISO/IEC Guide 2:2004*

standard uncertainty

uncertainty of the result of a measurement expressed as a standard deviation. *Ref JCGM 100:2008 2.3.1*

stereotype

defines how an existing UML metaclass may be extended as part of a *profile*.

subtype

secondary or subordinate type of categorization or classification of information.

table

two dimensional set of values given by a set of *key-value pairs table entries*.

table cell

subdivision of a *table entry* representing a singular value.

table entry

subdivision of a *table* containing a set of *key-value pairs* representing *table cells*.

top level

element that represents the most significant physical or logical functions of a piece of equipment.

type

classification or categorization of information.

uncertainty

uncertainty (of measurement) parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand. *Ref JCGM 100:2008 2.2.3*

Note: Use of the term uncertainty refers to uncertainty of measurement.

upper limit

upper conformance boundary for a variable.

upper warning

upper boundary indicating increased concern and supervision may be required.

version

unique identifier of the administered item. *Ref ISO/IEC 11179-:2015*

2.2 Information Model Terms

Asset Information Model

information model that provides semantic models for *Assets*.

Device Information Model

information model that describes the physical and logical configuration for a piece of equipment and the data that may be reported by that equipment.

Error Information Model

information model that describes the *response document* returned by an *agent* when it encounters an error while interpreting a *request* for information from a *client* or when an *agent* experiences an error while publishing the *response* to a *request* for information.

MTConnect Information Model

information model that defines the semantics of the MTConnect Standard.

Observation Information Model

information model that describes the *streaming data* reported by a piece of equipment.

2.3 Protocol Terms

asset request

HTTP Request to the *agent* regarding *Assets*.

current request

request to an *agent* to produce an *MTConnectStreams Response Document* containing the *Observation Information Model* for a snapshot of the latest observations at the moment of the *request* or at a given *sequence number*.

data streaming

method for an *agent* to provide a continuous stream of information in response to a single *request* from a *client*.

MTConnect Request

request for information issued from a *client* to an *MTConnect Agent*.

MTConnect Response Document

response document published by an *MTConnect Agent*.

MTConnectAssets Response Document

response document published by an *MTConnect Agent* in response to an *asset request*.

MTConnectDevices Response Document

response document published by an *MTConnect Agent* in response to a *probe request*.

MTConnectErrors Response Document

response document published by an *MTConnect Agent* whenever it encounters an error while interpreting an *MTConnect Request*.

MTConnectStreams Response Document

response document published by an *MTConnect Agent* in response to a *current request* or a *sample request*.

probe request

request to an *agent* to produce an *MTConnectDevices Response Document* containing the *Device Information Model*.

protocol

set of rules that allow two or more entities to transmit information from one to the other.

publish

sending of messages in a *publish and subscribe* pattern.

publish and subscribe

asynchronous communication method in which messages are exchanged between applications without knowing the identity of the sender or recipient.

> Note: In the MTConnect Standard, a communications messaging pattern that may be used to publish *streaming data* from an *agent*.

request

communications method where a *client* transmits a message to an *agent*. That message instructs the *agent* to respond with specific information.

request and response

communications pattern that supports the transfer of information between an *agent* and a *client*.

response

response *interface* which responds to a *request*.

sample request

request to an *agent* to produce an *MTConnectStreams Response Document* containing the *Observation Information Model* for a set of timestamped observations made by *Components*.

streaming data

observations published by a piece of equipment defined by the equipment metadata.

subscribe

receiving messages in a *publish and subscribe* pattern.

transport protocol

set of capabilities that provide the rules and procedures used to transport information between an *agent* and a client software application through a physical connection.

2.4 HTTP Terms

HTTP Body

data bytes transmitted in an HTTP transaction message immediately following the headers. *Ref IETF:RFC-2616*

HTTP Error Message

response provided by an *agent* indicating that an *HTTP Request* is incorrectly formatted or identifies that the requested data is not available from the *agent*. *Ref IETF:RFC-2616*

HTTP Header

header of either an *HTTP Request* from a *client* or an *HTTP Response* from an *agent*. *Ref IETF:RFC-2616*

HTTP Header Field

components of the header section of request and response messages in an HTTP transaction. *Ref IETF:RFC-2616*

HTTP Message

consist of requests from client to server and responses from server to client. *Ref IETF:RFC-2616*

> Note: In MTConnect Standard, it describes the information that is exchanged between an *agent* and a *client*.

HTTP Messaging

interface for information exchange functionality. *Ref IETF:RFC-2616*

HTTP Method

portion of a command in an *HTTP Request* that indicates the desired action to be performed on the identified resource; often referred to as verbs. *Ref IETF:RFC-2616*

HTTP Query

portion of a request for information that more precisely defines the specific information to be published in response to the request. *Ref IETF:RFC-2616*

HTTP Request

request message from a client to a server includes, within the first line of that message, the method to be applied to the resource, the identifier of the resource, and the protocol version in use. *Ref IETF:RFC-2616*

Note: In MTConnect Standard, a request issued by a *client* to an *agent* requesting information defined in the *HTTP Request Line*.

HTTP Request Line

begins with a method token, followed by the Request-URI and the protocol version, and ending with CRLF. A CRLF is allowed in the definition of TEXT only as part of a header field continuation. *Ref IETF:RFC-2616*

Note: the first line of an *HTTP Request* describing a specific *response document* to be published by an *agent*.

HTTP Request Method

indicates the method to be performed on the resource identified by the Request-URI. *Ref IETF:RFC-2616*

HTTP Request URI

Uniform Resource Identifier that identifies the resource upon which to apply the request. *Ref IETF:RFC-2616*

HTTP Response

after receiving and interpreting a request message, a server responds with an HTTP response message. *Ref IETF:RFC-2616*

Note: In MTConnect Standard, the information published from an *agent* in reply to an *HTTP Request*.

HTTP Server

server that accepts *HTTP Request* from *client* and publishes *HTTP Response* as a reply to those *HTTP Request*. *Ref IETF:RFC-2616*

HTTP Status Code

3-digit integer result code of the attempt to understand and satisfy the request. *Ref IETF:RFC-2616*

HTTP Version

version of the HTTP protocol. *Ref IETF:RFC-2616*

2.5 XML Terms

abstract element

element that defines a set of common characteristics that are shared by a group of elements. An abstract entity cannot appear in a document. In a specific implementation, an abstract entity is replaced by a derived element that is itself not an abstract entity. The characteristics for the derived element are inherited from the abstract entity.

attribute

additional information or property for an *element*.

child element

element of a data modeling structure that illustrates the relationship between itself and the higher-level *parent element* within which it is contained.

document body

portion of the content of an *MTConnect Response Document* that is defined by the relative *MTConnect Information Model*. The *document body* contains the *structural elements* and *Observations* or *DataItems* reported in a *response document*.

document header

portion of the content of an *MTConnect Response Document* that provides information from an *agent* defining version information, storage capacity, protocol, and other information associated with the management of the data stored in or retrieved from the *agent*.

element name

descriptive identifier contained in both the `start-tag` and `end-tag` of an XML element that provides the name of the element.

namespace

organizes information into logical groups.

parent element

element of a data modeling structure that illustrates the relationship between itself and the lower-level *child element*.

root element

first *structural element* provided in a *response document* encoded using XML.

structural element

element that organizes information that represents the physical and logical parts and sub-parts of a piece of equipment.

XML Document

structured text file encoded using Extensible Markup Language (XML).

XML Schema

schema defining a specific document encoded in XML.

2.6 MTConnect Terms

Asset

asset that is used by the manufacturing process to perform tasks.

Note 1 to entry: An *Asset* relies upon an *Device* to provide observations and information about itself and the *Device* revises the information to reflect changes to the *Asset* during their interaction. Examples of *Assets* are cutting tools, Part Information, Manufacturing Processes, Fixtures, and Files.

Note 2 to entry: A singular `assetId` uniquely identifies an *Asset* throughout its lifecycle and is used to track and relate the *Asset* to other *Devices* and entities.

Note 3 to entry: *Assets* are temporally associated with a device and can be removed from the device without damage or alteration to its primary functions.

Component

engineered system part of a *Device* composed of zero or more *Components*

Composition

Component belonging to a *Component* and not composed of any *Components*.

Configuration

configuration for a *Component*

DataItem

observable observed by a *Component* that may make *Observations*

Device

Component not belonging to any *Component* that may have assets

MTConnect Agent

agent for the *MTConnect Information Model*.

MTConnect Document

document that represents a Part(s) of the MTConnect Standard.

MTConnect Event

observation of either a state or discrete value of the *Component*.

MTConnect Interface

interaction model for interoperability between pieces of equipment.

Observation

observation that provides telemetry data for a *DataItem*.

2.7 Acronyms

2D

two-dimensional

3D

three-dimensional

AI

artificial intelligence

ALM

application lifecycle management

AMT

The Association for Manufacturing Technology

ANSI

American National Standards Institute

AP

Application Protocol

API

application programming interface

ASME

American Society of Mechanical Engineers

ASTM

American Society for Testing and Materials

AWS

American Welding Society

BDD

block definition diagram

BOM

bill of materials

BST

Board on Standardization and Testing

C&R

cause and remedy

CA

certificate authority

CAD

computer-aided design

CAE

computer-aided engineering

CAI

computer-aided inspection

CAM

computer-aided manufacturing

CAx

computer-aided technologies

CDATA

Character Data

CFD

computational fluid dynamics

CM

configuration management

CMS

coordinate-measurement system

CNC

Computer Numerical Controller

CNRI

Corporation for National Research Initiatives

CPM

Core Product Model

CPM2

Revised Core Product Model

CPSC

Consumer Product Safety Commission

cUAV

configurable unmanned aerial vehicle

DARPA

Defense Advanced Research Projects Agency

DER

designated-engineering representative

DFM

design for manufacturing

DLA

Defense Logistics Agency

DMC

digital manufacturing certificate

DMSC

Dimensional Metrology Standards Consortium

DNS

Domain Name System

DoD

U.S. Department of Defense

DOI

Distributed Object Identifier

DRM

digital rights management

ECR

engineering change request

ERP

enterprise resource planning

FAA

Federal Aviation Administration

FAIR

first article inspection reporting

FDA

Food and Drug Administration

FEA

finite-element analysis

GD&T

geometric dimensions and tolerances

GID

global identifier

HMI

Human Machine Interface

HTML

Hypertext Markup Language

HTTP

Hypertext Transfer Protocol

HTTPS

Hypertext Transfer Protocol over Secure Sockets Layer

I/O

in-out

ID

identifier

IEEE

Institute of Electrical and Electronics Engineers

IIoT

industrial internet of things

INCOSE

International Council on Systems Engineering

IP

intellectual property

ISO

International Standards Organization

ISS

International Space Station

ISV

Independent Software Vendor

IT

information technology

ITU-T

Telecommunication Standardization Sector of the International Telecommunication Union

JSON

JavaScript Object Notation

JT

Jupiter Tesselation

LHS

Lifecycle Handler System

LIFT

Lifecycle Information Framework and Technology

LOI

Lifecycle Object Identifier

MAC

media access control

MADE

Manufacturing Automation and Design Engineering

MBD

model-based definition

MBE

Model-Based Enterprise

MBI

model-based inspection

MBM

model-based manufacturing

MBSD

model-based standards development

MBSE

model-based systems engineering

MEDALS

Military Engineering Data Asset Locator System

MES

manufacturing execution system

MOI

manufacturing object identifier

MOM

Message Orienged Middleware

MQTT

Message Queuing Telemetry Transport

MTC

Manufacturing Technology Centre

NASA

National Aeronautics and Space Administration

NC

numerical control

NIST

National Institute of Standards and Technology

NMTOKEN

Name Token

NNMI

National Network of Manufacturing Innovation

NSF

National Science Foundation

NTSC

National Transportation Safety Board

OASIS

Organization for the Advancement of Structured Information Standards

ODI

Open Data Institute

OEM

original equipment manufacturer

OOI

Ocean Observatories Initiative

OPC

OLE for Process Control

OSLC

Open Services for Lifecycle Collaboration

OSTP

Office of Science and Technology Policy

OT

operational technology

OWL

Ontology Web Language

PDF

Portable Document Format

PDM

product-data management

PDQ

product-data quality

PHM

prognosis and health monitoring

PI

principal investigator

PLC

Programmable Logic Controller

PLCS

Product Life Cycle Support

PLM

product lifecycle management

PLOT

product lifecycle of trust

PMI

product and manufacturing information

PMS

Production Management System

PRC

Product Representation Compact

PSI

Physical Science Informatics

PTAB

Primary Trustworthy Digital Repository Authorization Body Ltd.

QIF

Quality Information Framework

QMS

quality management system

QName

Qualified Name

RDF

Resource Description Framework

REST

Representational State Transfer

RII

receiving and incoming inspection

S/MIME

Secure/Multipurpose Internet Mail Extensions

SaaS

software-as-a-service

SAML

Security Assertion Markup Language

SC

Standards Committee

SCADA

Supervisory Control And Data Acquisition

SDO

Standards Development Organization

SFTP

Secure File Transfer Protocol

SKOS

Simple Knowledge Organization System

SLH

system lifecycle handler

SLR

systematic literature review

SME

small-to-medium enterprise

SMOPAC

Smart Manufacturing Operations Planning and Control

SMS Test Bed

Smart Manufacturing Systems Test Bed

SOA

service-oriented architecture

SPMM

semantic-based product metamodel

SSL

Secure Sockets Layer

STEP

Standard for the Exchange of Product Model Data

STEP AP242

Standard for the Exchange of Product Model Data Application Protocol 242

STL

Stereolithography

SysML

Systems Modeling Language

TCP/IP

Transmission Control Protocol/Internet Protocol

TDP

technical data package

TLS

Transport Layer Security

TSM

Total System Model

UA

Unified Architecture

UAL

Unified Architecture Language

UML
: Unified Modeling Language

URI
: Uniform Resource Identifier

URL
: Uniform Resource Locator

URN
: Uniform Resource Name

UTC
: Coordinated Universal Time

UUID
: Universally Unique Identifier

V&V
: verification and validation

W3C
: World Wide Web Consortium

WSN
: Wirth Syntax Notation

WWW
: World Wide Web

X.509-PKI
: Public Key Infrastructure

X.509-PMI
: Privilege Management Infrastructure

XML
: Extensible Markup Language

XPath
: XML Path Language

XSD
: XML Schema Definitions

2.8 MTConnect References

[MTConnect Part 1.0] *MTConnect Standard Part 1.0 - Fundamentals*. Version 2.0.

[MTConnect Part 2.0] *MTConnect Standard: Part 2.0 - Device Information Model*. Version 2.0.

[MTConnect Part 3.0] *MTConnect Standard: Part 3.0 - Observation Information Model*. Version 2.0.

[MTConnect Part 4.0] *MTConnect Standard: Part 4.0 - Asset Information Model*. Version 2.0.

3 Device Information Model

The *Device Information Model* provides a representation of the physical and logical configuration for a piece of equipment used for a manufacturing process or for any other purpose. It also provides the definition of data that may be reported by that equipment.

Using information defined in the *Device Information Model*, a software application can determine the configuration and reporting capabilities of a piece of equipment. To do this, the software application issues a *probe request* (defined in *MTConnect Standard Part 1.0 - Fundamentals*) to an *agent* associated with a piece of equipment. An *agent* responds to the *probe request* with an *MTConnectDevices Response Document* that contains information describing both the physical and logical structure of the piece of equipment and a detailed description of each *Observation* that can be reported by the *agent* associated with the piece of equipment. This information allows the client software application to interpret the document and to extract the data with the same meaning, value, and context that it had at its original source.

The *MTConnectDevices Response Document* is comprised of two sections: `Header` and `Devices`.

The `Header` section contains protocol related information as defined in *MTConnect Standard Part 1.0 - Fundamentals*.

The `Devices` section of the *MTConnectDevices Response Document* contains a `Device` entity for each piece of equipment described in the document. Each `Device` is comprised of two primary types of entities - *Components* and *DataItems*.

Components organize information that represents the physical and logical parts and subparts of a piece of equipment (See *Section 4.1 - Components* for more details).

DataItems describe data that can be reported by a piece of equipment. In the *Device Information Model*, *DataItems* are defined as `DataItem` entities (See *Section 6.5 - DataItem Types*).

The *Components* and *DataItems* in the *MTConnectDevices Response Document* provide information representing the physical and logical structure for a piece of equipment and the types of data that the piece of equipment can report relative to that structure. The *MTConnectDevices Response Document* does not contain values for the data types reported by the piece of equipment. The *MTConnectStreams Response Document* defined in *MTConnect Standard: Part 3.0 - Observation Information Model* provides the data values that are reported by the piece of equipment.

Note: The MTConnect Standard also defines the information model for assets. An asset is something that is used in the manufacturing process, but is not permanently associated with a single piece of equipment, can be removed from the piece of equipment without compromising its function, and can be associated with other pieces of equipment during its lifecycle. See *MTConnect Standard: Part 4.0 - Asset Information Model* for more details on assets.

3.1 Device

`Component` composed of a piece of equipment that produces observations about itself.

A `Device` *organizes* its parts as `Component` entities.

A `Device` **MUST** have a `name` and `uuid` to identify itself.

A `Device` **MUST** have the following `DataItems`: `Availability`, `AssetChanged`, and `AssetRemoved`.

See *Section 4.1 - Components* for more details on the properties of `Device`.

See *Section 3.1.2 - Part Properties of Device* for a list of *top level* `Component` types for a `Device`.

3.1.1 Value Properties of Device

Table 1 lists the Value Properties of `Device`.

Value Property name	Value Property type	Multiplicity
`<<deprecated>> iso841Class`	`string`	0..1
`uuid`	`UUID`	1
`mtconnectVersion`	`string`	0..1
`name`	`string`	1
`hash`	`string`	0..1

Table 1: Value Properties of Device

Descriptions for Value Properties of `Device`:

- `<<deprecated>> iso841Class`

DEPRECATED in *MTConnect Version 1.2*.

- `mtconnectVersion`

  MTConnect version of the *Device Information Model* used to configure the information to be published for a piece of equipment in an *MTConnect Response Document*.

- `hash`

  condensed message digest from a secure one-way hash function. *Ref FIPS PUB 180-4*

3.1.2 Part Properties of Device

Table 2 lists the Part Properties of `Device`.

Part Property name	Multiplicity
`Auxiliary` (organized by `<<organizer>> Auxiliaries`)	0..*
`Controller` (organized by `<<organizer>> Controllers`)	0..*
`Interface` (organized by `<<organizer>> Interfaces`)	0..*
`Resource` (organized by `<<organizer>> Resources`)	0..*
`Structure` (organized by `<<organizer>> Structures`)	0..*
`System` (organized by `<<organizer>> Systems`)	0..*
`Axis` (organized by `<<organizer>> Axes`)	0..*
`Adapter` (organized by `<<organizer>> Adapters`)	0..*

Table 2: Part Properties of Device

Descriptions for Part Properties of `Device`:

- `Auxiliary`

  abstract `Component` composed of removable part(s) of a piece of equipment that provides supplementary or extended functionality.

- `Controller`

  `System` that provides regulation or management of a system or component. *Ref ISO 16484-5:2017*

- `Interface`

  abstract `Component` that coordinates actions and activities between pieces of equipment.

- `Resource`

  abstract `Component` composed of material or personnel involved in a manufacturing process.

- `Structure`

  `Component` composed of part(s) comprising the rigid bodies of the piece of equipment.

- `System`

  abstract `Component` that is permanently integrated into the piece of equipment.

- `Axis`

  abstract `Component` composed of a motion system that provides linear or rotational motion for a piece of equipment.

- `Adapter`

  `Component` that provides information about the data source for an *MTConnect Agent*.

3.1.3 Commonly Observed DataItem Types for Device

Table 3 lists the Commonly Observed DataItem Types for `Device`.

Commonly Observed DataItem Types	Multiplicity
`Availability`	1
`AssetChanged`	1
`AssetRemoved`	1

Table 3: Commonly Observed DataItem Types for Device

4 Components Model

MTConnectDevices provides the physical and logical architecture of a piece of equipment. Figure 1 provides an overview of the entities used in an example of an MTConnectDevices entity.

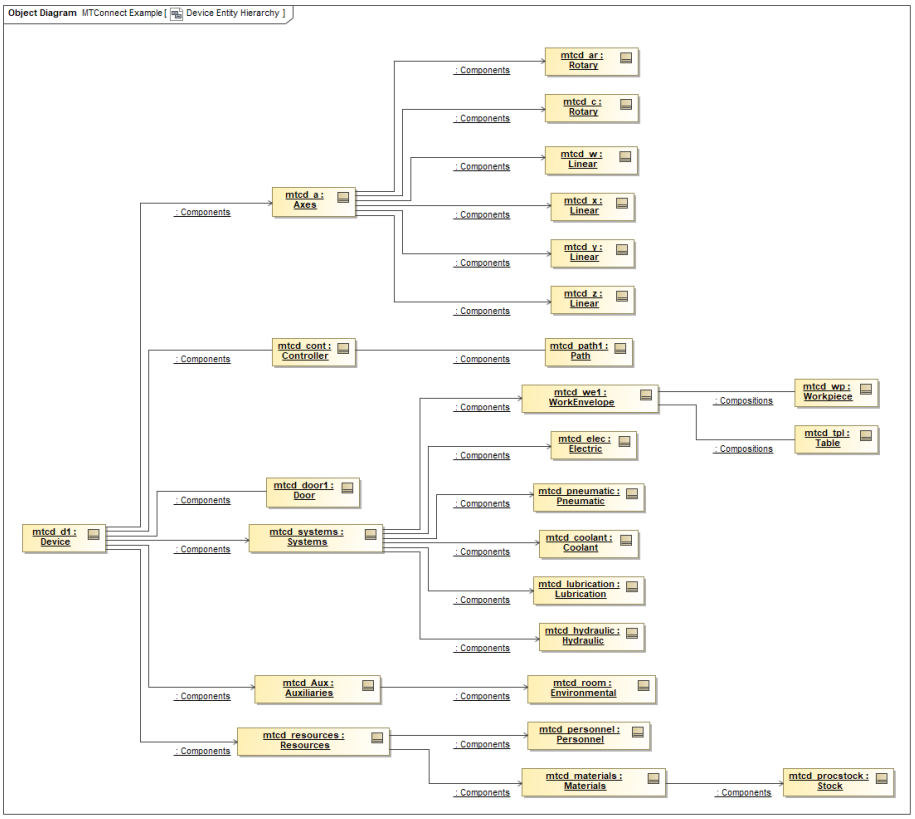

Figure 1: Device Entity Hierarchy Example

Note 1 to entry: See *Example 3* for the XML representation of the same example.

Note 2 to entry: Example above only shows the Component and Composition level hierarchy. For a complete semantics on each kind see *Section 4.1 - Components* and *Section 5.1 - Compositions*.

A variety of entities are defined to describe a piece of equipment. Some of these entities **MUST** always be defined for an MTConnectDevices entity, while others are optional and **MAY** be used, as required, to provide additional structure.

The first, or highest level, entity defined for an MTConnectDevices entity is Devices. Devices is used to group one or more pieces of equipment into a single document. Devices **MUST** always be defined for an MTConnectDevices entity.

Device is the next entity defined for the MTConnectDevices entity. A separate Device entity is used to identify each piece of equipment for an MTConnectDevices entity. Each Device provides information on the physical and logical structure of the piece of equipment and the data associated with that equipment. Device can also represent any logical grouping of pieces of equipment that function as a unit or any other data source that provides data through an *agent*.

One or more Device entities **MUST** always be defined for an MTConnectDevices entity.

Components is the next entity defined for the MTConnectDevices entity. Components is used to group information describing lower level physical parts or logical functions of a piece of equipment.

Component is the next level of entity defined for the MTConnectDevices entity. Component is both an abstract type entity and an *organizer* type entity.

As an abstract entity, Component will always be realized by a specific Component type defined in *Section 4.3 - Component Types*. Each Component can also be used to organize information describing *lower level* entities or *DataItems* associated with the Component.

If *lower level* entities are described, these entities are by definition child Component entities of a parent Component. At this next level, the *lower level* child Component entities are grouped by Components.

A Component **MAY** be further decomposed into Composition entities that are grouped by Compositions. These describe the lowest level basic structural or functional building blocks contained within a Component. Data provided for a Component provides more specific meaning when it is associated with one of the Composition entities of the Component. The different Composition types that **MAY** be defined for the MTConnectDevices entity are defined in *Section 5.1 - Compositions*.

This parent-child relationship can continue to any depth required to fully define a piece of equipment.

Note: See Figure 1 for an example.

4.1 Components

This section provides semantic information for the Component entity. Figure 2 shows the Component model.

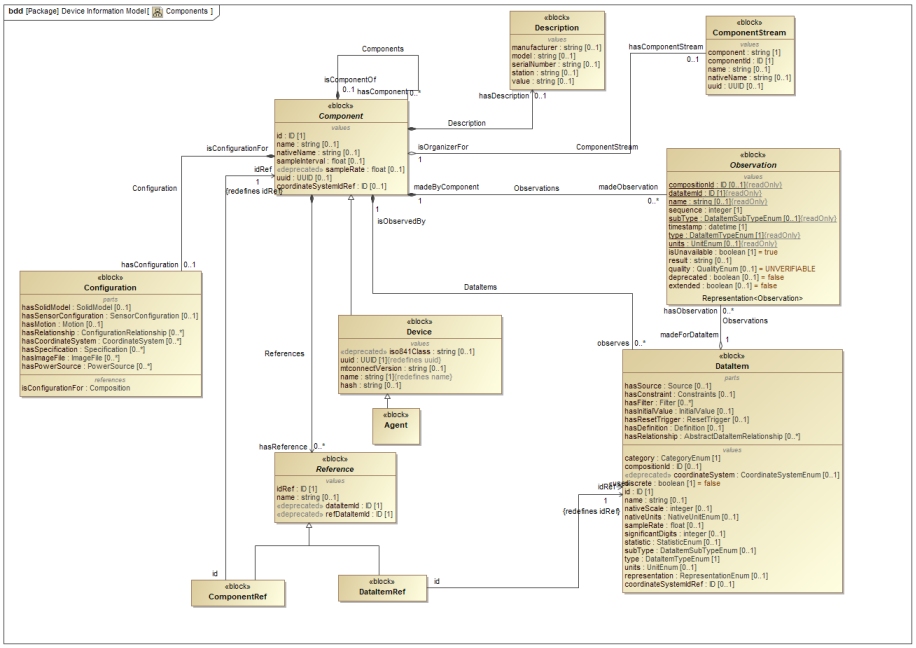

Figure 2: Components

Note: See *Section B.1 - Components Schema Diagrams* for XML schema.

4.1.1 Component

logical or physical entity that provides a capability.

Component is an abstract entity and will be realized by specific Component types for an MTConnectDevices entity. See *Section 4.3 - Component Types* for more details on the Component types.

Component also provides structure for describing the *lower level* entities associated with it.

At least one of Component, DataItem, or Reference entities **MUST** be provided for a Component.

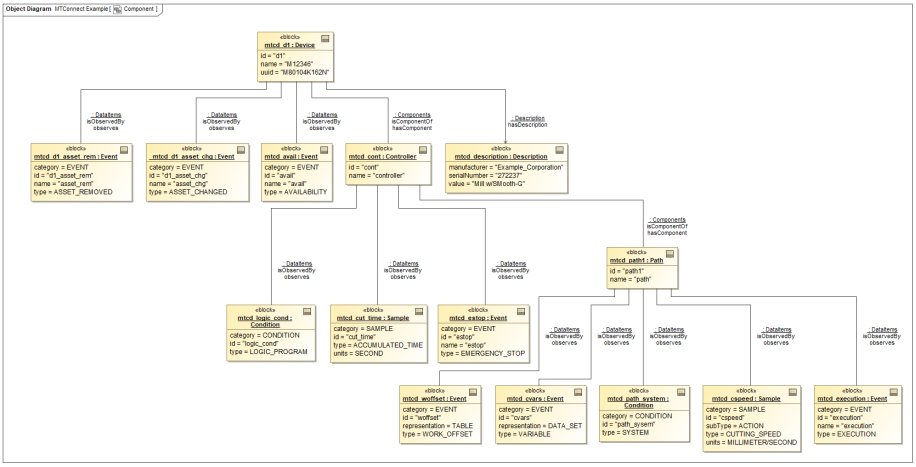

Figure 3: Component Example

Note: See *Example 4* for the XML representation of the same example.

4.1.1.1 Value Properties of Component

Table 4 lists the Value Properties of `Component`.

Value Property name	Value Property type	Multiplicity
`id`	`ID`	1
`name`	`string`	0..1
`nativeName`	`string`	0..1
`sampleInterval`	`float`	0..1
`<<deprecated>> sampleRate`	`float`	0..1
`uuid`	`UUID`	0..1
`coordinateSystemIdRef`	`ID`	0..1

Table 4: Value Properties of Component

Descriptions for Value Properties of `Component`:

- `id`

  unique identifier for the `Component`.

- `name`

  name of the `Component`.

  When provided, `name` **MUST** be unique for all child `Component` entities of a parent `Component`.

- `nativeName`

  common name associated with `Component`.

- `sampleInterval`

  interval in milliseconds between the completion of the reading of the data associated with the `Component` until the beginning of the next sampling of that data.

  This information may be used by client software applications to understand how often information from a `Component` is expected to be refreshed.

  The refresh rate for data from all child `Component` entities will be the same as for the parent `Component` element unless specifically overridden by another `sampleInterval` provided for the child `Component`.

- `<<deprecated>> sampleRate`

  DEPRECATED in *MTConnect Version 1.2*. Replaced by `sampleInterval`.

- `uuid`

  universally unique identifier for the `Component`.

- `coordinateSystemIdRef`

  specifies the `CoordinateSystem` for this `Component` and its children.

4.1.1.2 Reference Properties of Component

Table 5 lists the Reference Properties of `Component`.

Reference Property name	Multiplicity
`ComponentStream`	0..1

Table 5: Reference Properties of Component

Descriptions for Reference Properties of `Component`:

- `ComponentStream`

  organizes the data associated with each `Component` entity defined for a `Device` in the associated *MTConnectDevices Response Document*.

  `ComponentStream` *organizes* the `Observation` entities associated with the `Component`.

  See *MTConnect Standard: Part 3.0 - Observation Information Model* for the `ComponentStream` model.

Note 1 to entry: In the XML representation, `ComponentStream` entities **MUST NOT** appear in the *MTConnectDevices Response Document*.

Note 2 to entry: In the XML representation, `ComponentStream` entities **MUST** appear only in the *MTConnectStreams Response Document*.

4.1.1.3 Part Properties of Component

Table 6 lists the Part Properties of `Component`.

Part Property name	Multiplicity
`Description`	0..1
`Composition` (organized by `Compositions`)	0..*
`Component` (organized by `Components`)	0..*
`Configuration`	0..1
`DataItem` (organized by `DataItems`)	0..*
`Observation` (organized by `Observations`)	0..*
`Reference` (organized by `References`)	0..*

Table 6: Part Properties of Component

Descriptions for Part Properties of `Component`:

- `Description`

  descriptive content.

  See *Section 4.1.2 - Description*.

- `Composition`

  functional part of a piece of equipment contained within a `Component`.

  `Compositions` groups one or more `Composition` entities. See *Section 5.1 - Compositions*.

- `Component`

  logical or physical entity that provides a capability.

  `Components` groups one or more `Component` entities.

- `Configuration`

  technical information about an entity describing its physical layout, functional characteristics, and relationships with other entities.

  See *Section 8.1 - Configurations*.

- `DataItem`

  information reported about a piece of equipment.

  `DataItems` groups one or more `DataItem` entities. See *Section 6.1 - DataItems*.

- `Observation`

  abstract entity that provides telemetry data for a `DataItem` at a point in time.

  `Observations` groups one or more `Observations` made by the `Component` entity.

  `Component` make `Observations` about observed `DataItems`.

  See *MTConnect Standard: Part 3.0 - Observation Information Model* for the `Observation` model.

  Note 1 to entry: In the XML representation, `Observation` entities **MUST NOT** appear in the *MTConnectDevices Response Document*.

  Note 2 to entry: In the XML representation, `Observation` entities **MUST** appear only in the *MTConnectStreams Response Document*.

- `Reference`

  pointer to information that is associated with another entity defined elsewhere in the `MTConnectDevices` entity for a piece of equipment.

  `References` groups one or more `Reference` entities associated with the `Component`. See *Section 7.1 - References*.

4.1.2 Description

descriptive content.

Note 1 to entry: See Figure 3 for an example.

Note 2 to entry: See *Example 4* for the XML representation of the same example.

The value of `Description` **MUST** be `string`.

4.1.2.1 Value Properties of Description

Table 7 lists the Value Properties of `Description`.

Value Property name	Value Property type	Multiplicity
manufacturer	string	0..1
model	string	0..1
serialNumber	string	0..1
station	string	0..1

Table 7: Value Properties of Description

Descriptions for Value Properties of Description:

- manufacturer

  name of the manufacturer of the physical or logical part of a piece of equipment represented by this element.

- model

  model description of the physical part or logical function of a piece of equipment represented by this element.

- serialNumber

  serial number associated with a piece of equipment.

- station

  identifier where a manufacturing function takes place.

4.2 Devices

This section provides semantic information for the Device types.

4.2.1 Agent

Device composed of an *MTConnect Agent* and all its connected data sources.

An Agent **MUST** be provided by all *MTConnect Agent* implementations.

An Agent **MUST** provide notifications when devices are added or changed.

An Agent **MUST** provide connection information for each data source currently supplying data to the *MTConnect Agent*.

An `Agent` **MAY** provide information about telemetry relating to data sources.

An `Agent` **MAY** provide information about the *MTConnect Agent* resource utilization.

4.3 Component Types

This section provides semantic information for the types of `Component`.

Note: In the XML representation, `Component` entities are defined into two major categories:

- *top level* `Component` entities that *organizes* the most significant physical or logical functions of a piece of equipment (see *Section 3.1.2 - Part Properties of Device*). They **MAY** also be used as *lower level* `Component` entities; as required. See *Section 4.4 - Component Organizer Types*.

- *lower level* `Component` entities composed of the sub-parts of the parent `Component` to provide more clarity and granularity to the physical or logical structure of the *top level* `Component` entities.

This section provides guidance for the most common relationships between `Component` types. However, all `Component` types **MAY** be used in any configuration, as required, to fully describe a piece of equipment.

As described in *Section 4.1 - Components*, `Component` is an abstract entity and will be always realized by a specific `Component` type.

4.3.1 Adapter

`Component` that provides information about the data source for an *MTConnect Agent*.

It **MAY** contain connectivity state of the data source and additional telemetry about the data source and source-specific information.

4.3.2 Amplifier

leaf Component composed of an electronic component or circuit that amplifies power, electric current, or voltage.

4.3.2.1 Part Properties of Amplifier

Table 8 lists the Part Properties of Amplifier.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 8: Part Properties of Amplifier

Descriptions for Part Properties of Amplifier:

- Composition

  functional part of a piece of equipment contained within a Component.

- Component

  logical or physical entity that provides a capability.

4.3.3 Auxiliary

abstract Component composed of removable part(s) of a piece of equipment that provides supplementary or extended functionality.

4.3.3.1 Deposition

Auxiliary that manages the addition of material or state change of material being performed in an additive manufacturing process.

For example, this could describe the portion of a piece of equipment that manages a material extrusion process or a vat polymerization process.

4.3.3.2 Loader

`Auxiliary` that provides movement and distribution of materials, parts, tooling, and other items to or from a piece of equipment.

4.3.3.2.1 BarFeeder

`Loader` that delivers bar stock to a piece of equipment.

4.3.3.3 ToolingDelivery

`Auxiliary` that manages, positions, stores, and delivers tooling within a piece of equipment.

4.3.3.3.1 GangToolBar

`ToolingDelivery` composed of a tool mounting mechanism that holds any number of tools.

Tools are located in `Station` entities. Tools are positioned for use in the manufacturing process by linearly positioning the `GangToolBar`.

4.3.3.3.2 AutomaticToolChanger

`ToolingDelivery` composed of a tool delivery mechanism that moves tools between a `ToolMagazine` and a *spindle* a `Turret`.

`AutomaticToolChanger` may also transfer tools between a location outside of a piece of equipment and a `ToolMagazine` or `Turret`.

4.3.3.3.3 ToolMagazine

`ToolingDelivery` composed of a tool storage mechanism that holds any number of tools.

Tools are located in `Pots`. `Pots` are moved into position to transfer tools into or out of the `ToolMagazine` by an `AutomaticToolChanger`.

4.3.3.3.4 ToolRack

`ToolingDelivery` composed of a linear or matrixed tool storage mechanism that holds any number of tools.

Tools are located in `Station` entities.

4.3.3.3.5 Turret

`ToolingDelivery` composed of a tool mounting mechanism that holds any number of tools.

Tools are positioned for use in the manufacturing process by rotating the `Turret`.

4.3.3.4 WasteDisposal

`Auxiliary` that removes manufacturing byproducts from a piece of equipment.

4.3.4 Axis

abstract `Component` composed of a motion system that provides linear or rotational motion for a piece of equipment.

In robotics, the term *Axis* is synonymous with *Joint*. A *Joint* is the connection between two parts of the structure that move in relation to each other.

`Linear` and `Rotary` components **MUST** have `name` that **MUST** follow the conventions described below. Use the `nativeName` for the manufacturer's name of the axis if it differs from the assigned `name`.

MTConnect has two high-level classes for automation equipment as follows: (1) Equipment that controls cartesian coordinate axes and (2) Equipment that controls articulated axes. There are ambiguous cases where some machines exhibit both characteristics; when this occurs, the primary control system's configuration determines the classification.

Examples of cartesian coordinate equipment are CNC Machine Tools, Coordinate measurement machines, as specified in ISO 841, and 3D Printers. Examples of articulated automation equipment are Robotic systems as specified in ISO 8373.

The following sections define the designation of names for the axes and additional guid-

ance when selecting the correct scheme to use for a given piece of equipment.

4.3.4.1 Cartesian Coordinate Naming Conventions

A Three-Dimensional Cartesian Coordinate control system organizes its axes orthogonally relative to a machine coordinate system where the manufacturer of the equipment specifies the origin.

`Axes` `name` **SHOULD** comply with ISO 841, if possible.

4.3.4.1.1 Linear Motion

A piece of equipment **MUST** represent prismatic motion using a `Linear` axis and assign its `name` using the designations `X`, `Y`, and `Z`. A `Linear` axis `name` **MUST** append a monotonically increasing suffix when there are more than one parallel axes; for example, `X2`, `X3`, and `X4`.

4.3.4.1.2 Rotary Motion

MTConnect **MUST** assign the `name` to `Rotary` axes exhibiting rotary motion using `A`, `B`, and `C`. A `Rotary` axis `name` **MUST** append a monotonically increasing suffix when more than one `Rotary` axis rotates around the same `Linear` axis; for example, `A2`, `A3`, and `A4`.

4.3.4.2 Articulated Machine Control Systems

An articulated control system's axes represent the connecting linkages between two adjacent rigid members of an assembly. The `Linear` axis represents prismatic motion, and the `Rotary` axis represents the rotational motion of the two related members. The control organizes the axes in a kinematic chain from the mounting surface (base) to the end-effector or tooling.

4.3.4.3 Articulated Machine Axis Names

The axes of articulated machines represent forward kinematic relationships between mechanical linkages. Each axis is a connection between linkages, also referred to as joints, and **MUST** be named using a `J` followed by a monotonically increasing number; for example, `J1`, `J2`, `J3`. The numbering starts at the base axis connected or closest to the

mounting surface, J1, incrementing to the mechanical interface, Jn, where n is the number of the last axis. The chain forms a parent-child relationship with the parent being the axis closest to the base.

A machine having an axis with more than one child **MUST** number each branch using its numeric designation followed by a branch number and a monotonically increasing number. For example, if J2 has two children, the first child branch **MUST** be named J2.1.1 and the second child branch J2.2.1. A child of the first branch **MUST** be named J2.1.2, incrementing to J2.1.n, where J2.1.n is the number of the last axis in that branch.

4.3.4.4 Linear

Axis that provides prismatic motion along a fixed axis.

4.3.4.5 Part Properties of Linear

Table 9 lists the Part Properties of Linear.

Part Property name	Multiplicity
observesLoad	0..1
observesTemperature	0..1
observesAxisFeedrateActual	0..1

Table 9: Part Properties of Linear

Descriptions for Part Properties of Linear:

- Load
  actual versus the standard rating of a piece of equipment.
- Temperature
  degree of hotness or coldness measured on a definite scale.
- AxisFeedrate.Actual
  measured or reported value of an observation.

4.3.4.6 Rotary

Axis that provides rotation about a fixed axis.

4.3.4.7 Part Properties of Rotary

Table 10 lists the Part Properties of `Rotary`.

Part Property name	Multiplicity
observesLoad	0..1
observesTemperature	0..1
observesRotaryVelocity	0..1
observesAxisFeedrate	0..1

Table 10: Part Properties of Rotary

Descriptions for Part Properties of `Rotary`:

- `Load`
  actual versus the standard rating of a piece of equipment.
- `Temperature`
  degree of hotness or coldness measured on a definite scale.
- `RotaryVelocity`
  rotational speed of a rotary axis.
- `AxisFeedrate`
  feedrate of a linear axis.

4.3.4.8 <<deprecated>>Spindle

`Component` that provides an axis of rotation for the purpose of rapidly rotating a part or a tool to provide sufficient surface speed for cutting operations.

`Spindle` was **DEPRECATED** in *MTConnect Version 1.1* and was replaced by `RotaryMode`.

4.3.5 Ballscrew

leaf `Component` composed of a mechanical structure that transforms rotary motion into linear motion.

4.3.5.1 Part Properties of Ballscrew

Table 11 lists the Part Properties of `Ballscrew`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 11: Part Properties of Ballscrew

Descriptions for Part Properties of `Ballscrew`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.6 Belt

leaf `Component` composed of an endless flexible band that transmits motion for a piece of equipment or conveys materials and objects.

4.3.6.1 Part Properties of Belt

Table 12 lists the Part Properties of `Belt`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 12: Part Properties of Belt

Descriptions for Part Properties of `Belt`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.7 Brake

leaf Component that slows or stops a moving object by the absorption or transfer of the energy of momentum, usually by means of friction, electrical force, or magnetic force.

4.3.7.1 Part Properties of Brake

Table 13 lists the Part Properties of Brake.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 13: Part Properties of Brake

Descriptions for Part Properties of Brake:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.8 Chain

leaf Component composed of interconnected series of objects that band together and are used to transmit motion for a piece of equipment or to convey materials and objects.

4.3.8.1 Part Properties of Chain

Table 14 lists the Part Properties of Chain.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 14: Part Properties of Chain

Descriptions for Part Properties of Chain:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.9 Chopper

leaf Component that breaks material into smaller pieces.

4.3.9.1 Part Properties of Chopper

Table 15 lists the Part Properties of Chopper.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 15: Part Properties of Chopper

Descriptions for Part Properties of Chopper:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.10 Chuck

leaf Component composed of a mechanism that holds a part or stock material in place.

4.3.11 Chute

leaf Component composed of an inclined channel that conveys material.

4.3.11.1 Part Properties of Chute

Table 16 lists the Part Properties of Chute.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 16: Part Properties of Chute

Descriptions for Part Properties of Chute:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.12 CircuitBreaker

leaf Component that interrupts an electric circuit.

4.3.12.1 Part Properties of CircuitBreaker

Table 17 lists the Part Properties of CircuitBreaker.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 17: Part Properties of CircuitBreaker

Descriptions for Part Properties of CircuitBreaker:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.13 Clamp

leaf Component that strengthens, support, or fastens objects in place.

4.3.13.1 Part Properties of Clamp

Table 18 lists the Part Properties of Clamp.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 18: Part Properties of Clamp

Descriptions for Part Properties of Clamp:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.14 Compressor

leaf Component composed of a pump or other mechanism that reduces volume and increases pressure of gases in order to condense the gases to drive pneumatically powered pieces of equipment.

4.3.14.1 Part Properties of Compressor

Table 19 lists the Part Properties of Compressor.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 19: Part Properties of Compressor

Descriptions for Part Properties of Compressor:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.15 CoolingTower

leaf Component composed of a heat exchange system that uses a fluid to transfer heat to the atmosphere.

4.3.15.1 Part Properties of CoolingTower

Table 20 lists the Part Properties of CoolingTower.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 20: Part Properties of CoolingTower

Descriptions for Part Properties of CoolingTower:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.16 Door

Component composed of a mechanical mechanism or closure that can cover a physical access portal into a piece of equipment allowing or restricting access to other parts of the equipment.

The closure can be opened or closed to allow or restrict access to other parts of the equipment.

Door **MUST** have DoorState data item to indicate if the door is OPEN, CLOSED, or UNLATCHED. A Component **MAY** contain multiple Door entities.

4.3.16.1 Commonly Observed DataItem Types for Door

Table 21 lists the Commonly Observed DataItem Types for Door.

Commonly Observed DataItem Types	Multiplicity
DoorState	1

Table 21: Commonly Observed DataItem Types for Door

4.3.17 Drain

leaf Component that allows material to flow for the purpose of drainage from, for example, a vessel or tank.

4.3.17.1 Part Properties of Drain

Table 22 lists the Part Properties of Drain.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 22: Part Properties of Drain

Descriptions for Part Properties of Drain:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.18 Encoder

leaf Component that measures position.

4.3.18.1 Part Properties of Encoder

Table 23 lists the Part Properties of `Encoder`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 23: Part Properties of Encoder

Descriptions for Part Properties of `Encoder`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.19 Environmental

`Component` that observes the surroundings of another `Component`.

> Note: `Environmental` **SHOULD** be organized by `Auxillaries`, `Systems` or `Parts` depending on the relationship to the `Component`.

4.3.20 ExpiredPot

leaf `Component` that is a `Pot` for a tool that is no longer usable for removal from a `ToolMagazine` or `Turret`.

4.3.20.1 Part Properties of ExpiredPot

Table 24 lists the Part Properties of `ExpiredPot`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 24: Part Properties of ExpiredPot

Descriptions for Part Properties of ExpiredPot:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.21 ExposureUnit

leaf Component that emits a type of radiation.

4.3.21.1 Part Properties of ExposureUnit

Table 25 lists the Part Properties of ExposureUnit.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 25: Part Properties of ExposureUnit

Descriptions for Part Properties of ExposureUnit:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.22 ExtrusionUnit

leaf `Component` that dispenses liquid or powered materials.

4.3.22.1 Part Properties of ExtrusionUnit

Table 26 lists the Part Properties of `ExtrusionUnit`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 26: Part Properties of ExtrusionUnit

Descriptions for Part Properties of `ExtrusionUnit`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.23 Fan

leaf `Component` that produces a current of air.

4.3.23.1 Part Properties of Fan

Table 27 lists the Part Properties of `Fan`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 27: Part Properties of Fan

Descriptions for Part Properties of `Fan`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.24 Filter

leaf `Component` through which liquids or gases are passed to remove suspended impurities or to recover solids.

4.3.24.1 Part Properties of Filter

Table 28 lists the Part Properties of `Filter`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 28: Part Properties of Filter

Descriptions for Part Properties of `Filter`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.25 Galvanomotor

leaf `Component` composed of an electromechanical actuator that produces deflection of a beam of light or energy in response to electric current through its coil in a magnetic field.

4.3.25.1 Part Properties of Galvanomotor

Table 29 lists the Part Properties of `Galvanomotor`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 29: Part Properties of Galvanomotor

Descriptions for Part Properties of Galvanomotor:

- Composition

  functional part of a piece of equipment contained within a Component.

- Component

  logical or physical entity that provides a capability.

4.3.26 Gripper

leaf Component that holds a part, stock material, or any other item in place.

4.3.26.1 Part Properties of Gripper

Table 30 lists the Part Properties of Gripper.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 30: Part Properties of Gripper

Descriptions for Part Properties of Gripper:

- Composition

  functional part of a piece of equipment contained within a Component.

- Component

  logical or physical entity that provides a capability.

4.3.27 Hopper

leaf Component composed of a chamber or bin in which materials are stored temporarily, typically being filled through the top and dispensed through the bottom.

4.3.27.1 Part Properties of Hopper

Table 31 lists the Part Properties of Hopper.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 31: Part Properties of Hopper

Descriptions for Part Properties of Hopper:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.28 LinearPositionFeedback

leaf Component that measures linear motion or position.

DEPRECATION WARNING : May be deprecated in the future. Recommend using Encoder.

4.3.28.1 Part Properties of LinearPositionFeedback

Table 32 lists the Part Properties of LinearPositionFeedback.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 32: Part Properties of LinearPositionFeedback

Descriptions for Part Properties of `LinearPositionFeedback`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.29 Lock

`Component` that physically prohibits a `Device` or `Component` from opening or operating.

4.3.29.1 Commonly Observed DataItem Types for Lock

Table 33 lists the Commonly Observed DataItem Types for `Lock`.

Commonly Observed DataItem Types	Multiplicity
LockState	0..1

Table 33: Commonly Observed DataItem Types for Lock

4.3.30 Motor

leaf `Component` that converts electrical, pneumatic, or hydraulic energy into mechanical energy.

4.3.30.1 Part Properties of Motor

Table 34 lists the Part Properties of `Motor`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 34: Part Properties of Motor

Descriptions for Part Properties of Motor:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.31 Oil

leaf Component composed of a viscous liquid.

4.3.31.1 Part Properties of Oil

Table 35 lists the Part Properties of Oil.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 35: Part Properties of Oil

Descriptions for Part Properties of Oil:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.32 Part

abstract `Component` composed of a *part* being processed by a piece of equipment.

4.3.32.1 PartOccurrence

`Part` that exists at a specific place and time, such as a specific instance of a bracket at a specific timestamp.

`PartId` **MUST** be defined for `PartOccurrence`.

Example 1: XML Device Model Example for PartOccurrence and ComponentData

```
<Parts id="partOccSet">
  <Components>
    <PartOccurrence id="partOccur">
      <DataItems>
        <DataItem id="partSet" category="EVENT"
        representation="TABLE" type ="COMPONENT\
        textunderscore␣DATA">
          <Definition>
            <EntryDefinitions>
              <EntryDefinition keyType="
                PART\textunderscore␣
                UNIQUE\textunderscore␣ID"
                />
            </EntryDefinitions>
            <CellDefinitions>
              <CellDefinition key="
                partNumber" type="PART\
                textunderscore␣KIND\
                textunderscore␣ID"
                subType="PART\
                textunderscore␣NUMBER"/>
              <CellDefinition key="batchId
                " type="PART\
                textunderscore␣GROUP\
                textunderscore␣ID"
                subType="BATCH"/>
              <CellDefinition key="
                quantity" type="PART\
                textunderscore␣COUNT"
                subType="TARGET"/>
              <CellDefinition key="
                actualCompleteTime" type=
                "PROCESS\textunderscore␣
                TIME" subType="COMPLETE"
                />
```

```
                                        <CellDefinition key="
                                           partState" type="PROCESS\
                                           textunderscore␣STATE"/>
                                </CellDefinitions>
                            </Definition>
                        </DataItem>
                    </DataItems>
                </PartOccurrence>
            </Components>
</Parts>
```

Example 2: XML Streams Response Example for PartOccurrence and ComponentData

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="/styles/Streams.xsl"?>
<MTConnectStreams>
  <Streams>
    <DeviceStream name="VMC-3Axis" uuid="test\textunderscore␣27MAY">
      <ComponentStream component="PartOccurrence" name="partSet"
          componentId="partOccur">
        <Events>
          <ComponentDataTable dataItemId="partSet" timestamp="
     2020-10-28T19:45:43.070010Z" sequence="95" count="2">
            <Entry key="part1">
              <Cell key="actualStartTime">2009-06-15T00
                 :00:00.000000</Cell>
              <Cell key="partId">part1</Cell>
              <Cell key="partName">SomeName</Cell>
              <Cell key="uniqueID">abc-123</Cell>
            </Entry>
            <Entry key="part2">
              <Cell key="actualStartTime">2009-06-15T00
                 :00:00.007925</Cell>
              <Cell key="partId">part2</Cell>
              <Cell key="partName">AnotherName</Cell>
              <Cell key="uniqueID">def-123</Cell>
            </Entry>
          </ComponentDataTable>
        </Events>
      </ComponentStream>
    </DeviceStream>
  </Streams>
</MTConnectStreams>
```

4.3.32.2 Commonly Observed DataItem Types for PartOccurrence

Table 36 lists the Commonly Observed DataItem Types for `PartOccurrence`.

Commonly Observed DataItem Types	Multiplicity
PartId	1
PartUniqueId	0..1
PartGroupId	0..1
PartKindId	0..1
PartCount	0..1
PartStatus	0..1
ProcessOccurrenceId	0..1
ProcessTime	0..1
User	0..1

Table 36: Commonly Observed DataItem Types for PartOccurrence

4.3.32.2.1 FeatureOccurrence

Component that provides information related to an individual *feature*.

4.3.32.3 Part Properties of FeatureOccurrence

Table 37 lists the Part Properties of FeatureOccurrence.

Part Property name	Multiplicity
observesFeaturePersisitentId	0..1
observesFeatureMeasurement	0..1
observesMeasurementType	0..1
observesCharacteristicPersistentId	0..1
observesCharacteristicStatus	0..1
observesUncertainty	0..1
observesUncertaintyType	0..1
observesMeasurementUnits	0..1
observesMeasurementValue	0..1

Table 37: Part Properties of FeatureOccurrence

Descriptions for Part Properties of FeatureOccurrence:

- FeaturePersisitentId
  Universally Unique Identifier (UUID) of a *feature*. *Ref ISO 10303 AP 242/239*.
- FeatureMeasurement

tabular representation of assessing elements of a *feature*.

`FeatureMeasurement` **MAY** include a *characteristic* in which case it **MAY** include a `CHARACTERISTIC_STATUS`.

- `MeasurementType`

  class of measurement being performed. *Ref QIF 3:2018 Section 6.3*

  Examples: `POINT`, `RADIUS`, `ANGLE`, `LENGTH`, etc.

- `CharacteristicPersistentId`

  UUID of the *characteristic*.

- `CharacteristicStatus`

  pass/fail result of the measurement.

- `Uncertainty`

  uncertainty specified by `UncertaintyType`.

- `UncertaintyType`

  method used to compute *standard uncertainty*.

- `MeasurementUnits`

  engineering units of the measurement.

- `MeasurementValue`

  measurement based on the measurement type.

4.3.33 Path

`Component` that organizes an independent operation or function within a `Controller`.

For many types of equipment, `Path` organizes a set of `Axes`, one or more Program elements, and the data associated with the motion of a control point as it moves through space. However, it **MAY** also represent any independent function within a `Controller` that has unique data associated with that function.

`Path` **SHOULD** provide an `Execution` data item to define the operational state of the `Controller` of the piece of equipment.

If the `Controller` is capable of performing more than one independent operation or function simultaneously, a separate `Path` **MUST** be used to organize the data associated with each independent operation or function.

4.3.33.1 Part Properties of Path

Table 38 lists the Part Properties of `Path`.

Part Property name	Multiplicity
observesProgram	0..1
observesPathFeedrateOverrideProgrammed	0..1
observesPathFeedrateOverrideRapid	0..1
observesRotaryVelocityOverride	0..1
observesPathFeedrate	0..1
observesPartCount	0..1

Table 38: Part Properties of Path

Descriptions for Part Properties of `Path`:

- `Program`

  name of the logic or motion program being executed by the `Controller` component.

- `PathFeedrateOverride.Programmed`

  directive value without offsets and adjustments.

- `PathFeedrateOverride.Rapid`

  performing an operation faster or in less time than nominal rate.

- `RotaryVelocityOverride`

  percentage change to the velocity of the programmed velocity for a `Rotary` axis.

  This command represents a percentage change to the velocity calculated by a logic or motion program or set by a switch for a `Rotary` type axis.

- `PathFeedrate`

  feedrate for the axes, or a single axis, associated with a `Path` component.

- `PartCount`

  aggregate count of parts.

4.3.33.2 Commonly Observed DataItem Types for Path

Table 39 lists the Commonly Observed DataItem Types for `Path`.

Commonly Observed DataItem Types	Multiplicity
Execution	0..1

Table 39: Commonly Observed DataItem Types for Path

4.3.34 Pot

leaf Component composed of a tool storage location associated with a ToolMagazine or AutomaticToolChanger.

4.3.34.1 Part Properties of Pot

Table 40 lists the Part Properties of Pot.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 40: Part Properties of Pot

Descriptions for Part Properties of Pot:

- Composition

  functional part of a piece of equipment contained within a Component.

- Component

  logical or physical entity that provides a capability.

4.3.35 <<deprecated>>Power

Power was **DEPRECATED** in *MTConnect Version 1.1* and was replaced by Availability data item type.

4.3.36 PowerSupply

leaf Component that provides power to electric mechanisms.

4.3.36.1 Part Properties of PowerSupply

Table 41 lists the Part Properties of `PowerSupply`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 41: Part Properties of PowerSupply

Descriptions for Part Properties of `PowerSupply`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.37 Process

abstract `Component` composed of a manufacturing process being executed on a piece of equipment.

4.3.37.1 ProcessOccurrence

`Process` that takes place at a specific place and time, such as a specific instance of part-milling occurring at a specific timestamp.

`ProcessOccurrenceId` **MUST** be defined for `ProcessOccurrence`.

4.3.37.2 Commonly Observed DataItem Types for ProcessOccurrence

Table 42 lists the Commonly Observed DataItem Types for `ProcessOccurrence`.

Commonly Observed DataItem Types	Multiplicity
ProcessOccurrenceId	1
ProcessAggregateId	0..1
ProcessTime	0..1
ProcessKindId	0..1
User	0..1
Program	0..1
PartUniqueId	0..1

Table 42: Commonly Observed DataItem Types for ProcessOccurrence

4.3.38 Pulley

leaf Component composed of a mechanism or wheel that turns in a frame or block and serves to change the direction of or to transmit force.

4.3.38.1 Part Properties of Pulley

Table 43 lists the Part Properties of Pulley.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 43: Part Properties of Pulley

Descriptions for Part Properties of Pulley:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.39 Pump

leaf Component that raises, drives, exhausts, or compresses fluids or gases by means of a piston, plunger, or set of rotating vanes.

4.3.39.1 Part Properties of Pump

Table 44 lists the Part Properties of Pump.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 44: Part Properties of Pump

Descriptions for Part Properties of Pump:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.40 Reel

leaf Component composed of a rotary storage unit for material.

4.3.40.1 Part Properties of Reel

Table 45 lists the Part Properties of Reel.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 45: Part Properties of Reel

Descriptions for Part Properties of Reel:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.41 RemovalPot

leaf Component that is a Pot for a tool that has to be removed from a ToolMagazine or Turret to a location outside of the piece of equipment.

4.3.41.1 Part Properties of RemovalPot

Table 46 lists the Part Properties of RemovalPot.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 46: Part Properties of RemovalPot

Descriptions for Part Properties of RemovalPot:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.42 Resource

abstract Component composed of material or personnel involved in a manufacturing process.

4.3.42.1 Material

Resource composed of material that is consumed or used by the piece of equipment for production of parts, materials, or other types of goods.

4.3.42.1.1 Stock

Material that is used in a manufacturing process and to which work is applied in a machine or piece of equipment to produce parts.

Stock may be either a continuous piece of material from which multiple parts may be produced or it may be a discrete piece of material that will be made into a part or a set of parts.

4.3.42.2 Part Properties of Stock

Table 47 lists the Part Properties of Stock.

Part Property name	Multiplicity
observesMaterial	0..1
observesLengthRemaining	0..1
observesLengthStandard	0..1

Table 47: Part Properties of Stock

Descriptions for Part Properties of Stock:

- Material
  identifier of a material used or consumed in the manufacturing process.
- Length.Remaining
  remaining total length of an object.
- Length.Standard
  standard or original length of an object.

4.3.42.3 Personnel

Resource composed of an individual or individuals who either control, support, or otherwise interface with a piece of equipment.

4.3.42.4 Part Properties of Personnel

Table 48 lists the Part Properties of Personnel.

Part Property name	Multiplicity
observesUserOperator	0..1
observesUserMaintenance	0..1

Table 48: Part Properties of Personnel

Descriptions for Part Properties of Personnel:

- User.Operator

  identifier of the person currently responsible for operating the piece of equipment.

- User.Maintenance

  identifier of the person currently responsible for performing maintenance on the piece of equipment.

4.3.43 ReturnPot

leaf Component that is a Pot for a tool that has been removed from *spindle* or Turret and awaiting for return to a ToolMagazine.

4.3.43.1 Part Properties of ReturnPot

Table 49 lists the Part Properties of ReturnPot.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 49: Part Properties of ReturnPot

Descriptions for Part Properties of ReturnPot:

- Composition

  functional part of a piece of equipment contained within a Component.

- Component

  logical or physical entity that provides a capability.

4.3.44 SensingElement

leaf Component that provides a signal or measured value.

4.3.44.1 Part Properties of SensingElement

Table 50 lists the Part Properties of SensingElement.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 50: Part Properties of SensingElement

Descriptions for Part Properties of SensingElement:

- Composition

  functional part of a piece of equipment contained within a Component.

- Component

  logical or physical entity that provides a capability.

4.3.45 Sensor

Component that responds to a physical stimulus and transmits a resulting impulse or value from a sensing unit.

If modeling individual sensors, then sensor should be associated with the Component that the measured value is most closely associated.

When modeled as an Auxiliary, sensor **SHOULD** represent an integrated sensor unit system that provides signal processing, conversion, and communications. A sensor unit may have multiple *sensing elements*.

See SensorConfiguration for more details on the use and configuration of a Sensor.

4.3.45.1 <<deprecated>>Thermostat

Component composed of a sensor or an instrument that measures temperature.

Thermostat was **DEPRECATED** in *MTConnect Version 1.2* and was replaced by Temperature.

4.3.45.2 <<deprecated>>Vibration

Component composed of a sensor or an instrument that measures the amount and/or frequency of vibration within a system.

Vibration was **DEPRECATED** in *MTConnect Version 1.2* and was replaced by Displacement, Frequency etc.

4.3.46 Spreader

leaf Component that flattens or spreading materials.

4.3.46.1 Part Properties of Spreader

Table 51 lists the Part Properties of Spreader.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 51: Part Properties of Spreader

Descriptions for Part Properties of Spreader:

- Composition

  functional part of a piece of equipment contained within a Component.

- Component

  logical or physical entity that provides a capability.

4.3.47 StagingPot

leaf `Component` that is a `Pot` for a tool that is awaiting transfer to a `ToolMagazine` or `Turret` from outside of the piece of equipment.

4.3.47.1 Part Properties of StagingPot

Table 52 lists the Part Properties of `StagingPot`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 52: Part Properties of StagingPot

Descriptions for Part Properties of `StagingPot`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.48 Station

leaf `Component` composed of a storage or mounting location for a tool associated with a `Turret`, `GangToolBar`, or `ToolRack`.

4.3.48.1 Part Properties of Station

Table 53 lists the Part Properties of `Station`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 53: Part Properties of Station

Descriptions for Part Properties of `Station`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.49 StorageBattery

leaf `Component` composed of one or more cells in which chemical energy is converted into electricity and used as a source of power.

4.3.49.1 Part Properties of StorageBattery

Table 54 lists the Part Properties of `StorageBattery`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 54: Part Properties of StorageBattery

Descriptions for Part Properties of `StorageBattery`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.50 Structure

`Component` composed of part(s) comprising the rigid bodies of the piece of equipment.

4.3.50.1 Link

`Structure` that provides a connection between `Component` entities.

4.3.51 Switch

leaf Component that turns on or off an electric current or makes or breaks a circuit.

4.3.51.1 Part Properties of Switch

Table 55 lists the Part Properties of Switch.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 55: Part Properties of Switch

Descriptions for Part Properties of Switch:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.52 System

abstract Component that is permanently integrated into the piece of equipment.

4.3.52.1 Actuator

Component composed of a physical apparatus that moves or controls a mechanism or system.

It takes energy usually provided by air, electric current, or liquid and converts the energy into some kind of motion.

4.3.52.1.1 Hydraulic

System that provides movement and distribution of pressurized liquid throughout the piece of equipment.

4.3.52.2 Part Properties of Hydraulic

Table 56 lists the Part Properties of `Hydraulic`.

Part Property name	Multiplicity
`observesPressure`	0..1

Table 56: Part Properties of Hydraulic

Descriptions for Part Properties of `Hydraulic`:

- `Pressure`

  force per unit area measured relative to atmospheric pressure.

  Commonly referred to as gauge pressure.

4.3.52.2.1 Pneumatic

`System` that uses compressed gasses to actuate components or do work within the piece of equipment.

Note: Actuation is usually performed using a cylinder.

4.3.52.3 Part Properties of Pneumatic

Table 57 lists the Part Properties of `Pneumatic`.

Part Property name	Multiplicity
`observesPressure`	0..1

Table 57: Part Properties of Pneumatic

Descriptions for Part Properties of `Pneumatic`:

- `Pressure`

  force per unit area measured relative to atmospheric pressure.

  Commonly referred to as gauge pressure.

4.3.52.4 Controller

`System` that provides regulation or management of a system or component. *Ref ISO 16484-5:2017*

Typical types of controllers for a piece of equipment include CNC (Computer Numerical Control), PAC (Programmable Automation Control), IPC (Industrialized Computer), or IC (Imbedded Computer).

Note: In XML representation, `Controller` is a *top level* element.

4.3.52.5 Part Properties of Controller

Table 58 lists the Part Properties of `Controller`.

Part Property name	Multiplicity
`Path`	0..*
`observesEmergencyStop`	0..1
`observesSystemCondition`	0..1
`observesControllerMode`	0..1
`observesCommunicationsCondition`	0..1
`observesLogicProgramCondition`	0..1

Table 58: Part Properties of Controller

Descriptions for Part Properties of `Controller`:

- `Path`

  `Component` that organizes an independent operation or function within a `Controller`.

- `EmergencyStop`

  state of the emergency stop signal for a piece of equipment, controller path, or any other component or subsystem of a piece of equipment.

- `System`

  general purpose indication associated with an electronic component of a piece of equipment or a controller that represents a fault that is not associated with the operator, program, or hardware.

- `ControllerMode`

  current mode of the `Controller` component.

- `Communications`

  indication that the piece of equipment has experienced a communications failure.

- `LogicProgram`

  indication that an error occurred in the logic program or programmable logic controller (PLC) associated with a piece of equipment.

4.3.52.6 Coolant

`System` that provides distribution and management of fluids that remove heat from a piece of equipment.

4.3.52.7 Part Properties of Coolant

Table 59 lists the Part Properties of `Coolant`.

Part Property name	Multiplicity
`observesConcentration`	0..1

Table 59: Part Properties of Coolant

Descriptions for Part Properties of `Coolant`:

- `Concentration`

  percentage of one component within a mixture of components.

4.3.52.8 Dielectric

`System` that manages a chemical mixture used in a manufacturing process being performed at that piece of equipment.

For example, this could describe the dielectric system for an EDM process or the chemical bath used in a plating process.

4.3.52.9 Electric

System composed of the main power supply for the piece of equipment that provides distribution of that power throughout the equipment.

The electric system will provide all the data with regard to electric current, voltage, frequency, etc. that applies to the piece of equipment as a functional unit. Data regarding electric power that is specific to a Component will be reported for that specific Component.

4.3.52.10 Part Properties of Electric

Table 60 lists the Part Properties of Electric.

Part Property name	Multiplicity
observesWattage	0..1

Table 60: Part Properties of Electric

Descriptions for Part Properties of Electric:

- Wattage

  power flowing through or dissipated by an electrical circuit or piece of equipment.

4.3.52.11 Enclosure

System composed of a structure that is used to contain or isolate a piece of equipment or area.

Enclosure may provide information regarding access to the internal components of a piece of equipment or the conditions within the enclosure. For example, Door may be defined as a *lower level* Component or Composition entity of the Enclosure.

4.3.52.12 EndEffector

System composed of functions that form the last link segment of a piece of equipment.

It is the part of a piece of equipment that interacts with the manufacturing process.

4.3.52.13 Feeder

`System` that manages the delivery of materials within a piece of equipment.

For example, this could describe the wire delivery system for an EDM or welding process; conveying system or pump and valve system distributing material to a blending station; or a fuel delivery system feeding a furnace.

4.3.52.14 Lubrication

`System` that provides distribution and management of fluids used to lubricate portions of the piece of equipment.

4.3.52.15 ProcessPower

`System` composed of a power source associated with a piece of equipment that supplies energy to the manufacturing process separate from the `Electric` system.

For example, this could be the power source for an EDM machining process, an electroplating line, or a welding system.

4.3.52.16 Protective

`System` that provides functions used to detect or prevent harm or damage to equipment or personnel.

`Protective` does not include the information relating to the `Enclosure`.

4.3.52.17 Heating

`System` that delivers controlled amounts of heat to achieve a target temperature at a specified heating rate.

> Note: As an example, Energy Delivery Method can be either through Electric heaters or Gas burners.

4.3.52.18 Vacuum

`System` that evacuates gases and liquids from an enclosed and sealed space to a controlled negative pressure or a molecular density below the prevailing atmospheric level.

4.3.52.19 Cooling

`System` that extracts controlled amounts of heat to achieve a target temperature at a specified cooling rate.

> Note: As an example, Energy Extraction Method can be via cooling water pipes running through the chamber.

4.3.52.20 Pressure

`System` that delivers compressed gas or fluid and controls the pressure and rate of pressure change to a desired target set-point.

> Note: For example, Delivery Method can be a Compressed Air or N2 tank that is piped via an inlet valve to the chamber.

4.3.52.21 WorkEnvelope

`System` composed of the physical process execution space within a piece of equipment.

`WorkEnvelope` **MAY** provide information regarding the physical workspace and the conditions within that workspace.

4.3.52.22 AirHandler

system that circulates air or regulates airflow without altering temperature or humidity.

4.3.53 Table

leaf `Component` composed of a surface for holding an object or material.

4.3.53.1 Part Properties of Table

Table 61 lists the Part Properties of `Table`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 61: Part Properties of Table

Descriptions for Part Properties of `Table`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.54 Tank

leaf `Component` generally composed of an enclosed container.

4.3.54.1 Part Properties of Tank

Table 62 lists the Part Properties of `Tank`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 62: Part Properties of Tank

Descriptions for Part Properties of `Tank`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.55 Tensioner

leaf Component that provides or applies a stretch or strain to another mechanism.

4.3.55.1 Part Properties of Tensioner

Table 63 lists the Part Properties of Tensioner.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 63: Part Properties of Tensioner

Descriptions for Part Properties of Tensioner:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.56 TransferArm

leaf Component that physically moves a tool from one location to another.

4.3.56.1 Part Properties of TransferArm

Table 64 lists the Part Properties of TransferArm.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 64: Part Properties of TransferArm

Descriptions for Part Properties of TransferArm:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.57 TransferPot

leaf `Component` that is a `Pot` for a tool that is awaiting transfer from a `ToolMagazine` to *spindle* or `Turret`.

4.3.57.1 Part Properties of TransferPot

Table 65 lists the Part Properties of `TransferPot`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 65: Part Properties of TransferPot

Descriptions for Part Properties of `TransferPot`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.58 Transformer

leaf `Component` that transforms electric energy from a source to a secondary circuit.

4.3.58.1 Part Properties of Transformer

Table 66 lists the Part Properties of `Transformer`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 66: Part Properties of Transformer

Descriptions for Part Properties of `Transformer`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.59 Valve

leaf `Component` that halts or controls the flow of a liquid, gas, or other material through a passage, pipe, inlet, or outlet.

4.3.59.1 Part Properties of Valve

Table 67 lists the Part Properties of `Valve`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 67: Part Properties of Valve

Descriptions for Part Properties of `Valve`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.60 Vat

leaf `Component` generally composed of an open container.

4.3.60.1 Part Properties of Vat

Table 68 lists the Part Properties of `Vat`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 68: Part Properties of Vat

Descriptions for Part Properties of `Vat`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.3.61 Water

leaf `Component` composed of H_2O.

4.3.61.1 Part Properties of Water

Table 69 lists the Part Properties of `Water`.

Part Property name	Multiplicity
`hasComposition`	0
`hasComponent`	0

Table 69: Part Properties of Water

Descriptions for Part Properties of `Water`:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.62 Wire

leaf Component composed of a string like piece or filament of relatively rigid or flexible material provided in a variety of diameters.

4.3.62.1 Part Properties of Wire

Table 70 lists the Part Properties of Wire.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 70: Part Properties of Wire

Descriptions for Part Properties of Wire:

- Composition
  functional part of a piece of equipment contained within a Component.
- Component
  logical or physical entity that provides a capability.

4.3.63 Workpiece

leaf Component composed of an object or material on which a form of work is performed.

4.3.63.1 Part Properties of Workpiece

Table 71 lists the Part Properties of Workpiece.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 71: Part Properties of Workpiece

Descriptions for Part Properties of `Workpiece`:

- `Composition`
  functional part of a piece of equipment contained within a `Component`.
- `Component`
  logical or physical entity that provides a capability.

4.4 Component Organizer Types

This section provides semantic information for the types of `Component` that are used to *organize* other `Component` types.

4.4.1 <<organizer>>Adapters

`Component` that *organizes* `Adapter` types.

4.4.2 <<organizer>>Auxiliaries

`Component` that *organizes* `Auxiliary` types.

4.4.3 <<organizer>>Axes

`Component` that *organizes* `Axis` types.

4.4.4 <<organizer>>Controllers

`Component` that *organizes* `Controller` entities.

4.4.5 <<organizer>>Interfaces

Component that *organizes* Interface types.

4.4.6 <<organizer>>Parts

Component that *organizes* Part types.

4.4.7 <<organizer>>Processes

Component that *organizes* Process types.

4.4.8 <<organizer>>Resources

Component that *organizes* Resource types.

4.4.8.1 <<organizer>>Materials

Resources that *organizes* Material types.

4.4.9 <<organizer>>Structures

Component that *organizes* Structure types.

4.4.10 <<organizer>>Systems

Component that *organizes* System types.

5 Compositions Model

Composition entities are used to describe the lowest level physical building blocks of a piece of equipment contained within a Component. By referencing a specific Composition entity, further clarification and meaning to data associated with a specific Component can be achieved.

Both Component and Composition entities are *lower level* entities representing the sub-parts of the parent Component. However, there are distinct differences between Component and Composition type entities.

Component entities may be further defined with *lower level* Component entities and may have associated *DataItems*.

Composition entities represent the lowest level physical part of a piece of equipment. They **MUST NOT** be further defined with *lower level* Component entities and they **MUST NOT** have *DataItems* directly associated with them. They do provide additional information that can be used to enhance the specificity of *DataItems* associated with the parent Component.

5.1 Compositions

This section provides semantic information for the Composition entity.

See Figure 2 for the Composition model diagram.

Note: See *Example 5* for the XML representation of the same example.

5.1.1 Composition

functional part of a piece of equipment contained within a Component.

Composition **MUST NOT** have child Component, Composition, or DataItems elements.

5.1.1.1 Value Properties of Composition

Table 72 lists the Value Properties of Composition.

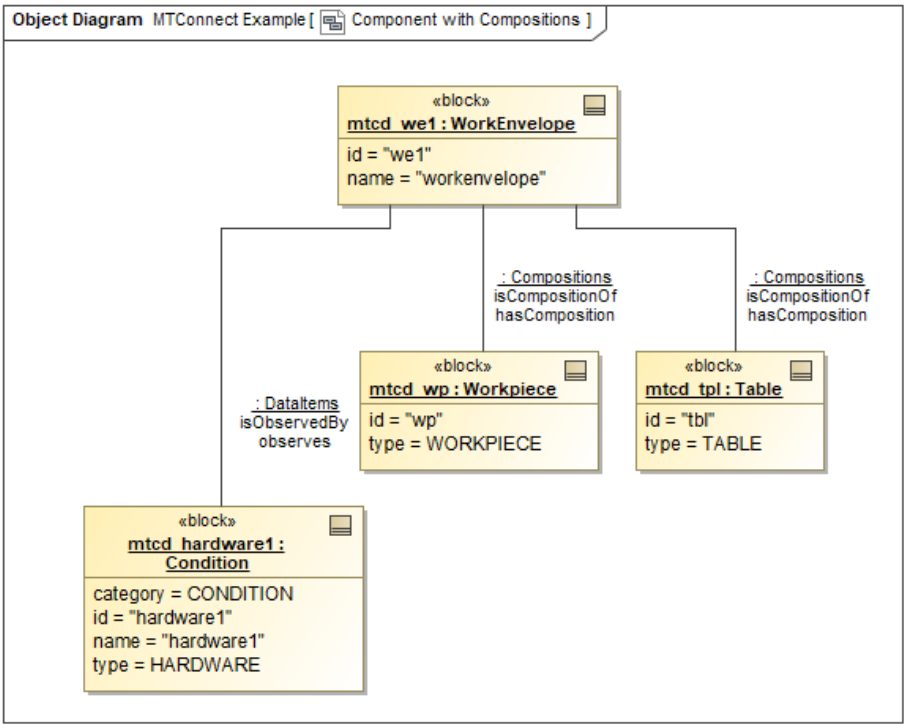

Figure 4: Component with Compositions Example

Value Property name	Value Property type	Multiplicity
`id`	`ID`	1
`type`	`CompositionTypeEnum`	1
`name`	`string`	0..1
`uuid`	`UUID`	0..1

Table 72: Value Properties of Composition

Descriptions for Value Properties of `Composition`:

- `id`

  unique identifier for the `Composition` element.

- `type`

  type of `Composition`.

  The value of `type` **MUST** be one of the `CompositionTypeEnum` enumeration.

- `name`

  name of the `Composition` element.

- `uuid`

  universally unique identifier for the `Composition`.

5.1.1.2 Part Properties of Composition

Table 73 lists the Part Properties of `Composition`.

Part Property name	Multiplicity
`Description`	0..1
`Configuration`	0..1

Table 73: Part Properties of Composition

Descriptions for Part Properties of `Composition`:

- `Description`

  descriptive content.

  See *Section 4.1.2 - Description*.

- `Configuration`

  technical information about an entity describing its physical layout, functional characteristics, and relationships with other entities.

  See *Section 8 - Configurations Model*.

6 DataItems Model

For an MTConnectDevices entity, *DataItems* describe data that can be reported by a piece of equipment and are associated with Device and Component entities. While the *DataItems* describe the data that can be reported by a piece of equipment as an MTConnectDevices entity, the actual data values are provided by the MTConnectStreams entity in the *MTConnectStreams Response Document*. See *MTConnect Standard: Part 3.0 - Observation Information Model* for detail on the reported values.

Each *DataItem* **SHOULD** be modeled for the MTConnectDevices entity such that it is associated with the entity that the reported data directly applies.

DataItem describes specific types of *DataItems* that represent a numeric value, a functioning state, or a health status reported by a piece of equipment. DataItem provides a detailed description for each *DataItem* that is reported; it defines the type of data being reported and an array of optional attributes that further describe that data. The different types of DataItem elements are defined in *Section 6.5 - DataItem Types*.

6.1 DataItems

This section provides semantic information for the DataItem entity. Figure 5 shows the DataItem model.

Note: See *Section B.2 - DataItems Schema Diagrams* for XML schema.

6.1.1 DataItem

information reported about a piece of equipment.

6.1.1.1 Value Properties of DataItem

Table 74 lists the Value Properties of DataItem.

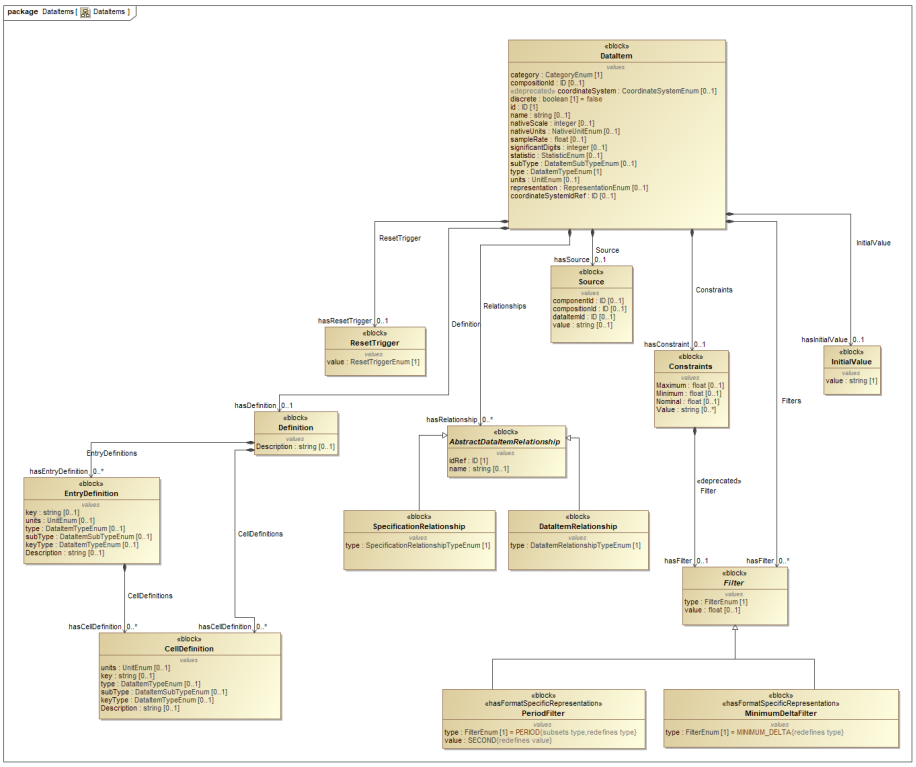

Figure 5: DataItems

Value Property name	Value Property type	Multiplicity
category	CategoryEnum	1
compositionId	ID	0..1
<<deprecated>> coordinateSystem	CoordinateSystemEnum	0..1
discrete	boolean	1
id	ID	1
name	string	0..1
nativeScale	integer	0..1
nativeUnits	NativeUnitEnum	0..1
sampleRate	float	0..1
significantDigits	integer	0..1
statistic	StatisticEnum	0..1
subType	DataItemSubTypeEnum	0..1
type	DataItemTypeEnum	1
units	UnitEnum	0..1
representation	RepresentationEnum	0..1
coordinateSystemIdRef	ID	0..1

Table 74: Value Properties of DataItem

Descriptions for Value Properties of `DataItem`:

- `category`

  specifies the kind of information provided by a data item.

  `CategoryEnum` Enumeration:

  - `CONDITION`

    information about the health of a piece of equipment and its ability to function.
  - `EVENT`

    discrete piece of information from the piece of equipment.
  - `SAMPLE`

    continuously variable or analog data value.

    A continuous value can be measured at any point-in-time and will always produce a result.

- `compositionId`

  identifier attribute of the `Composition` that the reported data is most closely associated.

- `<<deprecated>> coordinateSystem`

  for measured values relative to a coordinate system like `Position`, the coordinate system used may be reported.

  DEPRECATED in *Version 2.0*. Replaced by `coordinateSystemIdRef`.

- `discrete`

  indication signifying whether each value reported for the *Observation* is significant and whether duplicate values are to be suppressed.

  If a value is not defined for `discrete`, the default value **MUST** be `false`.

- `id`

  unique identifier for this data item.

- `name`

  name of the data item.

- `nativeScale`

  used to convert the reported value to represent the original measured value.

- `nativeUnits`

  native units of measurement for the reported value of the data item.

  `<<extensible>> NativeUnitEnum` Enumeration:

  - `AMPERE_HOUR`
    electric charge in ampere hour.
  - `BAR`
    pressure in bar.
  - `CENTIPOISE`
    viscosity in centipoise.
  - `CUBIC_FOOT`
    geometric volume in cubic foot.
  - `CUBIC_FOOT/HOUR`
    change of geometric volume in cubic foot per hour.
  - `CUBIC_FOOT/MINUTE`
    change of geometric volume in cubic foot per minute.
  - `DEGREE/MINUTE`
    rotational velocity in degree per minute.
  - `FAHRENHEIT`
    temperature in Fahrenheit.
  - `FOOT`
    length in foot.
  - `FOOT/MINUTE`
    speed in foot per minute.
  - `FOOT/SECOND`
    speed in foot per second.
  - `FOOT/SECOND`^{2}
    acceleration in foot per second squared.
  - `FOOT_3D`
    point in space identified by X, Y, and Z positions and represented by a space-delimited set of numbers each expressed in feet.
  - `GALLON/MINUTE`
    volumetric flow in gallon per minute.
  - `GRAVITATIONAL_ACCELERATION`
    acceleration relative to earth's gravity given in meter per second squared.

Note 1 to entry: At different points on Earth's surface, the free-fall acceleration ranges from 9.764 to 9.834 m/s2 (Wikipedia: Gravitational Acceleration). The constant can be customized depending on the location in the universe.

Note 2 to entry: In the standard, it is assumed that Earth's average value of gravitational acceleration is 9.90665 m/s2.

- `GRAVITATIONAL_FORCE`
  `MASS` times `GRAVITATIONAL_ACCELERATION` (g).
- `HOUR`
  time in hour.
- `INCH`
  length in inch.
- `INCH/MINUTE`
  speed in inch per minute.
- `INCH/REVOLUTION`
  feedrate per revolution in inch per revolution.
- `INCH/SECOND`
  speed in inch per second.
- `INCH/SECOND`^{2}
  acceleration in inch per second squared.
- `INCH_3D`
  point in space identified by X, Y, and Z positions and represented by a space-delimited set of numbers each expressed in inches.
- `INCH_POUND`
  torque in inch pound.
- `KELVIN`
  temperature in Kelvin.
- `KILOWATT`
  power in kilowatt.
- `KILOWATT_HOUR`
  energy in kilowatt-hour.
- `LITER/MINUTE`
  volumetric flow in liter per minute.
- `MICROMETER`
  length in micrometer.

- MICROTORR
  pressure in microtorr.
- MILLIMETER/MINUTE
  speed in millimeter per minute.
- MILLIMETER_MERCURY
  pressure in millimeter of mercury (mmHg).
- MINUTE
  time in minute.
- OTHER
  unsupported unit.
- PASCAL/MINUTE
  pressurization rate in pascal per minute.
- POUND
  mass in pound.
- POUND/INCH^2
  pressure in pound per square inch (PSI).
- RADIAN
  angle in radian.
- RADIAN/MINUTE
  angular velocity in radian per minute.
- RADIAN/SECOND
  angular velocity in radian per second.
- RADIAN/SECOND^2
  angular acceleration in radian per second squared.
- RANKINE
  temperature in Rankine.
- SQUARE_INCH
  geometric area in inch squared.
- TORR
  pressure in torr.

- sampleRate
  rate at which successive samples of a data item are recorded by a piece of equipment.
- significantDigits
  number of significant digits in the reported value.

- statistic

  type of statistical calculation performed on a series of data samples to provide the reported data value.

  StatisticEnum Enumeration:

  - AVERAGE

    mathematical average value calculated for the data item during the calculation period.

  - <<deprecated>> KURTOSIS

    DEPRECATED in *Version 1.6*. ∼∼A measure of the "peakedness" of a probability distribution; i.e., the shape of the distribution curve.∼∼

  - MAXIMUM

    maximum or peak value recorded for the data item during the calculation period.

  - MEDIAN

    middle number of a series of numbers.

  - MINIMUM

    minimum value recorded for the data item during the calculation period.

  - MODE

    number in a series of numbers that occurs most often.

  - RANGE

    difference between the maximum and minimum value of a data item during the calculation period. Also represents Peak-to-Peak measurement in a waveform.

  - ROOT_MEAN_SQUARE

    mathematical Root Mean Square (RMS) value calculated for the data item during the calculation period.

  - STANDARD_DEVIATION

    statistical Standard Deviation value calculated for the data item during the calculation period.

- subType

  sub-categorization of the type.

  <<extensible>> DataItemSubTypeEnum Enumeration:

  - ABORTED

    actions or activities that were attempted, but terminated before they could be completed.

- ABSOLUTE
  relating to or derived in the simplest manner from the fundamental units or measurements.
- ACTION
  indication of the operating state of a mechanism.
- ACTIVE
  relating to logic or motion program currently executing.
- ACTIVITY
  phase or segment of a recipe or program.
- ACTUAL
  reported value of an observation.
- ALL
  all actions, items, or activities being counted independent of the outcome.
- <<deprecated>> ALTERNATING
  measurement of alternating voltage or current. If not specified further in statistic, defaults to RMS voltage.
  DEPRECATED in *Version 1.6*.
- AUXILIARY
  when multiple locations on a piece of bar stock being feed by a bar feeder are referenced as the indication of whether the end of that piece of bar stock has been reached.
- A_SCALE
  A-Scale weighting factor on the frequency scale.
- BAD
  actions, items, or activities being counted that do not conform to specification or expectation.
- BATCH
  group of parts produced in a batch.
- BINARY
  observed as a binary data type.
- BOOLEAN
  observed as a boolean data type.
- BRINELL
  scale to measure the resistance to deformation of a surface.
- B_SCALE
  B-Scale weighting factor on the frequency scale.

- COMMANDED
  directive value including adjustments such as an offset or overrides.
- COMPLETE
  associated with the completion of an activity or event.
- CONSUMED
  amount of material consumed from an object or container during a manufacturing process.
- CONTROL
  state of the enabling signal or control logic that enables or disables the function or operation of the entity.
- C_SCALE
  C-Scale weighting factor on the frequency scale.
- DELAY
  elapsed time of a temporary halt of action.
- DETECT
  indicated by the presence or existence of something.
- <<deprecated>> DIRECT
  DC current or voltage.
  DEPRECATED in *Version 1.6*.
- DRY_RUN
  setting or operator selection used to execute a test mode to confirm the execution of machine functions.
- D_SCALE
  D-Scale weighting factor on the frequency scale.
- ENDED
  boundary when an activity or an event terminates.
- ENUMERATED
  observed as a set containing a restricted number of discrete values where each discrete value is named and unique. *Ref ISO 21961:2003, 013*
- EXPIRATION
  relating to the expiration or end of useful life for a material or other physical item.
- FAILED
  actions or activities that were attempted , but failed to complete or resulted in an unexpected or unacceptable outcome.

- FIRST_USE
  relating to the first use of a material or other physical item.
- GAS
  fluid that has no definite shape or volume.
- GATEWAY
  Gateway for the Component network.
- GOOD
  actions, items, or activities being counted that conform to specification or expectation.
- HEAT_TREAT
  material heat number.
- INCREMENTAL
  relating to or derived from the last observation.
- INSTALL_DATE
  date the hardware or software was installed.
- IPV4_ADDRESS
  IPV4 network address of the Component.
- IPV6_ADDRESS
  IPV6 network address of the Component.
- ISO_STEP_EXECUTABLE
  reference to a ISO 10303 Executable.
- JOG
  relating to momentary activation of a function or a movement.
  DEPRECATION WARNING: May be deprecated in the future.
- LATERAL
  indication of the position of a mechanism that may move in a lateral direction.
- LEEB
  scale to measure the elasticity of a surface.
- LENGTH
  reference to a length type tool offset variable.
- LICENSE
  license code to validate or activate the hardware or software.
- LINE
  state of the power source.

- LINEAR
  direction of motion of a linear motion.
- LIQUID
  fluid that has a definite volume but no definite shape.
- LOADED
  indication that the subparts of a piece of equipment are under load.
- LOT
  group of parts tracked as a lot.
- MACHINE_AXIS_LOCK
  setting or operator selection that changes the behavior of the controller on a piece of equipment.
- MAC_ADDRESS
  Media Access Control Address. The unique physical address of the network hardware.
- MAIN
  relating to the primary logic or motion program currently being executed.
- MAINTENANCE
  relating to maintenance on the piece of equipment.
- MANUAL_UNCLAMP
  indication of the state of an operator controlled interlock that can inhibit the ability to initiate an unclamp action of an electronically controlled chuck.
- MANUFACTURE
  related to the production of a material or other physical item.
- MANUFACTURER
  corporate identity for the maker of the hardware or software.
- MAXIMUM
  maximum value.
- MEASURED
  that has *uncertainty*.
- MINIMUM
  minimum value.
- MODEL
  model info of the hardware or software.
- MOHS
  scale to measure the resistance to scratching of a surface.

- `MOTION`
  indication of the open or closed state of a mechanism.
- `NO_SCALE`
  no weighting factor on the frequency scale.
- `OPERATING`
  piece of equipment that is powered or performing any activity.
- `OPERATION`
  step of a discrete manufacturing process.
- `OPERATOR`
  relating to the person currently responsible for operating the piece of equipment.
- `OPTIONAL_STOP`
  setting or operator selection that changes the behavior of the controller on a piece of equipment.
- `ORDER_NUMBER`
  authorization of a process occurrence.
- `OVERRIDE`
  overridden value.
- `PART`
  amount included in the *part*.
- `PART_FAMILY`
  group of parts having similarities in geometry, manufacturing process, and/or functions.
- `PART_NAME`
  word or set of words by which a part is known, addressed, or referred to.
- `PART_NUMBER`
  particular part design or model.
- `POWERED`
  piece of equipment is powered and functioning or `Component` that are required to remain on are powered.
- `PRIMARY`
  main or principle.
- `PROBE`
  position provided by a measurement probe.
  DEPRECATION WARNING: May be deprecated in the future.

- PROCESS
  relating to production of a part or product on a piece of equipment.
- PROCESS_NAME
  word or set of words by which a process being executed (process occurrence) by the device is known, addressed, or referred to.
- PROCESS_PLAN
  process plan that a process occurrence belongs to.
- PROCESS_STEP
  step in the process plan that this occurrence corresponds to.
- PROGRAMMED
  directive value without offsets and adjustments.
- RADIAL
  reference to a radial type tool offset variable.
- RAPID
  performing an operation faster or in less time than nominal rate.
- RAW_MATERIAL
  material that is used to produce parts.
- RECIPE
  process as part of product production; can be a subprocess of a larger process.
- RELEASE_DATE
  date the hardware or software was released for general use.
- REMAINING
  remaining measure or count of an action, object or activity.
- REQUEST
  request by an Interface for a task.
- RESPONSE
  response by an Interface to a *request* for a task.
- ROCKWELL
  scale to measure the resistance to deformation of a surface.
- ROTARY
  direction of a rotary motion using the right hand rule convention.
- SCHEDULE
  identity of a control program that is used to specify the order of execution of other programs.

- SEGMENT
  phase of a recipe process.
- SERIAL_NUMBER
  serial number that uniquely identifies a specific part.
- SET_UP
  relating to the preparation of a piece of equipment for production or restoring the piece of equipment to a neutral state after production.
- SHORE
  scale to measure the resistance to deformation of a surface.
- SINGLE_BLOCK
  setting or operator selection that changes the behavior of the controller on a piece of equipment.
- SOLID
  matter that has a definite shape and a definite volume.
- STANDARD
  standard measure of an object or an action.
- START
  boundary when an activity or an event commences.
- SUBNET_MASK
  SubNet mask for the Component network.
- SWITCHED
  indication of the activation state of a mechanism represented by a Composition.
- TARGET
  goal of the operation or process.
- TARGET_COMPLETION
  relating to the end or completion of an activity or event.
- TOOL_CHANGE_STOP
  setting or operator selection that changes the behavior of the controller on a piece of equipment.
- USEABLE
  remaining usable measure of an object or action.
- UUID
  universally unique identifier as specified in ISO 11578 or RFC 4122.

- VERSION
  version of the hardware or software.
- VERTICAL
  indication of the position of a mechanism that may move in a vertical direction.
- VICKERS
  scale to measure the resistance to deformation of a surface.
- VLAN_ID
  layer2 Virtual Local Network (VLAN) ID for the Component network.
- WASTE
  amount discarded.
- WIRELESS
  identifies whether the connection type is wireless.
- WORKING
  piece of equipment performing any activity, the equipment is active and performing a function under load or not.

- type
  type of data being measured. See *Section 6.5 - DataItem Types*.
  The value of type **MUST** be one of the DataItemTypeEnum enumeration.

- units
  unit of measurement for the reported value of the data item.
  <<extensible>> UnitEnum Enumeration:

  - AMPERE
    electric current in ampere.
  - CELSIUS
    temperature in degree Celsius.
  - COULOMB
    electric charge in coulomb.
  - COUNT
    count of something.
  - COUNT/SECOND
    frequency in count per second.
  - CUBIC_METER
    geometric volume in meter.

- CUBIC_MILLIMETER
  geometric volume in millimeter.
- CUBIC_MILLIMETER/SECOND
  change of geometric volume per second.
- CUBIC_MILLIMETER/SECOND^2
  change in geometric volume per second squared.
- DECIBEL
  sound level in decibel.
- DEGREE
  angle in degree.
- DEGREE/SECOND
  angular velocity in degree per second.
- DEGREE/SECOND^2
  angular acceleration in degree per second squared.
- DEGREE_3D
  space-delimited, floating-point representation of the angular rotation in degrees around the X, Y, and Z axes relative to a cartesian coordinate system respectively in order as A, B, and C.
  If any of the rotations is not known, it **MUST** be zero (0).
- GRAM
  mass in gram.
- GRAM/CUBIC_METER
  density in gram per cubic meter.
- HERTZ
  frequency in cycles per second.
- JOULE
  energy in joule.
- KILOGRAM
  mass in kilogram.
- LITER
  volume in liter.
- LITER/SECOND
  volumetric flow in liter per second.
- METER/SECOND^2
  acceleration in meter per second squared.

- MICRO_RADIAN
  tilt in micro radian.
- MILLIGRAM
  mass in milligram.
- MILLIGRAM/CUBIC_MILLIMETER
  density in milligram per cubic millimeter.
- MILLILITER
  volume in milliliter.
- MILLIMETER
  length in millimeter.
- MILLIMETER/REVOLUTION
  feedrate per revolution in millimeter per revolution.
- MILLIMETER/SECOND
  speed in millimeter per second.
- MILLIMETER/SECOND^2
  acceleration in millimeter per second squared.
- MILLIMETER_3D
  point in space identified by X, Y, and Z positions and represented by a space-delimited set of numbers each expressed in millimeters.
- NEWTON
  force in newton.
- NEWTON_METER
  torque in newton-meter.
- OHM
  electrical resistance in ohm.
- OHM_METER
  resistivity in ohm-meter.
- PASCAL
  pressure or stress in pascal.
- PASCAL/SECOND
  pressurization rate in pascal per second.
- PASCAL_SECOND
  viscosity in pascal-second.
- PERCENT
  amount in or for every hundred.

- PH
  acidity or alkalinity of a solution in pH.
- REVOLUTION/MINUTE
  rotational velocity in revolution per minute.
- REVOLUTION/SECOND
  rotational velocity in revolution per second.
- REVOLUTION/SECOND^2
  rotational acceleration in revolution per second squared.
- SECOND
  time in second.
- SIEMENS/METER
  electrical conductivity in siemens per meter.
- SQUARE_MILLIMETER
  geometric area in millimeter.
- UNIT_VECTOR_3D
  3D Unit Vector.
  Space delimited list of three floating point numbers.
- VOLT
  electric potential, electric potential difference or electromotive force in volt.
- VOLT_AMPERE
  apparent power in an electrical circuit, equal to the product of root-mean-square (RMS) voltage and RMS current (commonly referred to as VA) in volt-ampere.
- VOLT_AMPERE_REACTIVE
  reactive power in an AC electrical circuit (commonly referred to as VAR) in volt-ampere-reactive.
- WATT
  power in watt.
- WATT_SECOND
  electrical energy in watt-second

- representation

  description of a means to interpret data consisting of multiple data points or samples reported as a single value.

  If representation is not specified, it **MUST** be determined to be VALUE.

  RepresentationEnum Enumeration:

- DATA_SET

  reported value(s) are represented as a set of *key-value pairs*.

  Each reported value in the *data set* **MUST** have a unique key.

- <<deprecated>> DISCRETE

  DEPRECATED as representation type in *MTConnect Version 1.5*. Replaced by the discrete.

- TABLE

  two dimensional set of *key-value pairs* where the Entry represents a row, and the value is a set of *key-value pair* Cell elements.

  A *table* follows the same behavior as the *data set* for change tracking, clearing, and history. When an Entry changes, all Cell elements update as a single unit following the behavior of a *data set*.

  > Note: It is best to use Variable if the Cell entities represent multiple semantic types.

  Each Entry in the *table* **MUST** have a unique key. Each Cell of each Entry in the *table* **MUST** have a unique key.

  See Representation in *MTConnect Standard: Part 3.0 - Observation Information Model*, for a description of Entry and Cell elements.

- TIME_SERIES

  series of sampled data.

  The data is reported for a specified number of samples and each sample is reported with a fixed period.

- VALUE

  measured value of the sample data.

  If no representation is specified for a data item, the representation **MUST** be determined to be VALUE.

- coordinateSystemIdRef

  associated CoordinateSystem context for the DataItem.

6.1.1.2 Reference Properties of DataItem

Table 75 lists the Reference Properties of DataItem.

Reference Property name	Multiplicity
Observation (organized by Observations)	0..*

Table 75: Reference Properties of DataItem

Descriptions for Reference Properties of DataItem:

- Observation

  abstract entity that provides telemetry data for a DataItem at a point in time.

  Observations groups one or more Observation entities made for the DataItem entity.

  Component observes DataItem entities to create Observation entities for the DataItem entities.

  See *MTConnect Standard: Part 3.0 - Observation Information Model* for the Observation model.

  Note 1 to entry: In the XML representation, Observation entities **MUST NOT** appear in the *MTConnectDevices Response Document*.

  Note 2 to entry: In the XML representation, Observation entities **MUST** appear only in the *MTConnectStreams Response Document*.

6.1.1.3 Part Properties of DataItem

Table 76 lists the Part Properties of DataItem.

Part Property name	Multiplicity
Source	0..1
Constraints	0..1
Filter (organized by Filters)	0..*
InitialValue	0..1
ResetTrigger	0..1
Definition	0..1
AbstractDataItemRelationship (organized by Relationships)	0..*

Table 76: Part Properties of DataItem

Descriptions for Part Properties of DataItem:

- `Source`

  identifies the `Component`, `DataItem`, or `Composition` from which a measured value originates.

  See *Section 6.2.2 - Source*.

- `Constraints`

  organizes a set of expected values that can be reported for a `DataItem`.

  `Constraints` organizes a set of expected values that can be reported for the `DataItem`. See *Section 6.2.7 - Constraints*.

- `Filter`

  provides a means to control when an *agent* records updated information for a `DataItem`.

  `Filters` groups one or more `Filter` entities associated with the `DataItem`.

- `InitialValue`

  starting value for a `DataItem` as well as the value to be set for the `DataItem` after a reset event.

- `ResetTrigger`

  type of event that may cause a reset to occur.

- `Definition`

  defines the meaning of `Entry` and `Cell` entities when the associated `representation` is either `DATA_SET` or `TABLE`.

  See *Section 6.2.8 - Definition*.

- `AbstractDataItemRelationship`

  association between a `DataItem` and another entity.

  `Relationships` groups one or more `DataItemRelationship` and `SpecificationRelationship`. See *Section 8.4 - Relationships*.

6.2 Properties of DataItem

This section provides additional semantic information for the Part Properties of `DataItem`.

Note: See *Section B.2 - DataItems Schema Diagrams* for XML schema of the elements for `DataItem`.

6.2.1 ResetTrigger

type of event that may cause a reset to occur.

`ResetTriggerEnum` Enumeration:

- `ACTION_COMPLETE`

  observation of the `DataItem` that is measuring an action or operation is to be reset upon completion of that action or operation.

- `ANNUAL`

  observation of the `DataItem` is to be reset at the end of a 12-month period.

- `DAY`

  observation of the `DataItem` is to be reset at the end of a 24-hour period.

- `LIFE`

  observation of the `DataItem` is not reset and accumulates for the entire life of the piece of equipment.

- `MAINTENANCE`

  observation of the `DataItem` is to be reset upon completion of a maintenance event.

- `MONTH`

  observation of the `DataItem` is to be reset at the end of a monthly period.

- `POWER_ON`

  observation of the `DataItem` is to be reset when power was applied to the piece of equipment after a planned or unplanned interruption of power has occurred.

- `SHIFT`

  observation of the `DataItem` is to be reset at the end of a work shift.

- `WEEK`

  observation of the `DataItem` is to be reset at the end of a 7-day period.

6.2.2 Source

identifies the Component, DataItem, or Composition from which a measured value originates.

The value of Source **MUST** be string.

6.2.2.1 Value Properties of Source

Table 77 lists the Value Properties of Source.

Value Property name	Value Property type	Multiplicity
componentId	ID	0..1
compositionId	ID	0..1
dataItemId	ID	0..1

Table 77: Value Properties of Source

Descriptions for Value Properties of Source:

- componentId

  identifier of the Component that represents the physical part of a piece of equipment where the data represented by the DataItem originated.

- compositionId

  identifier of the Composition that represents the physical part of a piece of equipment where the data represented by the DataItem originated.

- dataItemId

  identifier of the DataItem that represents the originally measured value of the data referenced by this DataItem.

6.2.3 InitialValue

starting value for a DataItem as well as the value to be set for the DataItem after a reset event.

The value of InitialValue **MUST** be string.

6.2.4 Filter

provides a means to control when an *agent* records updated information for a `DataItem`.

The value of `Filter` **MUST** be `float`.

6.2.4.1 Value Properties of Filter

Table 78 lists the Value Properties of `Filter`.

Value Property name	Value Property type	Multiplicity
`type`	`FilterEnum`	1

Table 78: Value Properties of Filter

Descriptions for Value Properties of `Filter`:

- `type`

  type of `Filter`.

  `FilterEnum` Enumeration:

  - `MINIMUM_DELTA`

    new value **MUST NOT** be reported for a data item unless the measured value has changed from the last reported value by at least the delta given as the value of this element.

    The value of `Filter` **MUST** be an absolute value using the same units as the reported data.

  - `PERIOD`

    data reported for a data item is provided on a periodic basis. The `PERIOD` for reporting data is defined in the value of the `Filter`.

    The value of `Filter` **MUST** be an absolute value reported in seconds representing the time between reported samples of the value of the data item.

6.2.5 <<hasFormatSpecificRepresentation>>MinimumDeltaFilter

new value **MUST NOT** be reported for a data item unless the measured value has changed from the last reported value by at least the delta given as the value of this element.

The value of `Filter` **MUST** be an absolute value using the same units as the reported data.

6.2.6 <<hasFormatSpecificRepresentation>>PeriodFilter

data reported for a data item is provided on a periodic basis. The `PERIOD` for reporting data is defined in the value of the `Filter`.

The value of `Filter` **MUST** be an absolute value reported in seconds representing the time between reported samples of the value of the data item.

The value of `PeriodFilter` **MUST** be `SECOND`. See *Section 9.1.9 - SECOND*.

6.2.7 Constraints

organizes a set of expected values that can be reported for a `DataItem`.

6.2.7.1 Value Properties of Constraints

Table 79 lists the Value Properties of `Constraints`.

Value Property name	Value Property type	Multiplicity
`Maximum`	`float`	0..1
`Minimum`	`float`	0..1
`Nominal`	`float`	0..1
`Value`	`string`	0..*

Table 79: Value Properties of Constraints

Descriptions for Value Properties of `Constraints`:

- `Maximum`

  numeric upper constraint.

  If the data reported for a data item is a range of numeric values, the expected value reported **MAY** be described with an upper limit defined by this constraint.

- `Minimum`

numeric lower constraint.

If the data reported for a data item is a range of numeric values, the expected value reported **MAY** be described with a lower limit defined by this constraint.

- `Nominal`

  numeric target or expected value.

- `Value`

  single data value that is expected to be reported for a `DataItem`.

  `Value` **MUST NOT** be used in conjunction with any other `Constraint` elements.

6.2.7.2 Part Properties of Constraints

Table 80 lists the Part Properties of `Constraints`.

Part Property name	Multiplicity
`<<deprecated>> Filter`	0..1

Table 80: Part Properties of Constraints

Descriptions for Part Properties of `Constraints`:

- `Filter`

  provides a means to control when an *agent* records updated information for a `DataItem`.

  DEPRECATED in *MTConnect Version 1.4*. Moved to the `Filters`. See *Section 6.2 - Properties of DataItem*.

6.2.8 Definition

defines the meaning of `Entry` and `Cell` entities when the associated `representation` is either `DATA_SET` or `TABLE`.

6.2.8.1 Part Properties of Definition

Table 81 lists the Part Properties of `Definition`.

Part Property name	Multiplicity
CellDefinition (organized by CellDefinitions)	0..*
EntryDefinition (organized by EntryDefinitions)	0..*

Table 81: Part Properties of Definition

Descriptions for Part Properties of Definition:

- CellDefinition

  semantic definition of a Cell.

  CellDefinitions groups one or more CellDefinition entities. See *Section 6.3.1 - CellDefinition*.

- EntryDefinition

  semantic definition of an Entry.

  EntryDefinitions groups one or more EntryDefinition entities. See *Section 6.3.2 - EntryDefinition*.

6.3 Properties of Definition

This section provides semantic information for the elements of the Definition for a DataItem.

6.3.1 CellDefinition

semantic definition of a Cell.

6.3.1.1 Value Properties of CellDefinition

Table 82 lists the Value Properties of CellDefinition.

Value Property name	Value Property type	Multiplicity
units	UnitEnum	0..1
key	string	0..1
type	DataItemTypeEnum	0..1
subType	DataItemSubTypeEnum	0..1
keyType	DataItemTypeEnum	0..1

Table 82: Value Properties of CellDefinition

Descriptions for Value Properties of CellDefinition:

- units

  same as units. See *Section 6.1.1.1 - Value Properties of DataItem*.

  The value of units **MUST** be one of the UnitEnum enumeration.

- key

  unique identification of the Cell in the Definition.

  The description applies to all Cell observations having this key.

- type

  same as type. See *Section 6.5 - DataItem Types*.

  The value of type **MUST** be one of the DataItemTypeEnum enumeration.

- subType

  same as subType. See *Section 6.1.1 - DataItem*.

  The value of subType **MUST** be one of the DataItemSubTypeEnum enumeration.

- keyType

  type that defines the meaning of key.

  The value of keyType **MUST** be one of the DataItemTypeEnum enumeration.

6.3.2 EntryDefinition

semantic definition of an Entry.

6.3.2.1 Value Properties of EntryDefinition

Table 83 lists the Value Properties of `EntryDefinition`.

Value Property name	Value Property type	Multiplicity
`key`	`string`	0..1
`units`	`UnitEnum`	0..1
`type`	`DataItemTypeEnum`	0..1
`subType`	`DataItemSubTypeEnum`	0..1
`keyType`	`DataItemTypeEnum`	0..1

Table 83: Value Properties of EntryDefinition

Descriptions for Value Properties of `EntryDefinition`:

- `key`

  unique identification of the `Entry` in the `Definition`.

  The description applies to all `Entry` observations having this `key`.

- `units`

  same as `units`. See *Section 6.1.1.1 - Value Properties of DataItem*.

  The value of `units` **MUST** be one of the `UnitEnum` enumeration.

- `type`

  same as `type`. See *Section 6.5 - DataItem Types*.

  The value of `type` **MUST** be one of the `DataItemTypeEnum` enumeration.

- `subType`

  same as `subType`. See *Section 6.1.1 - DataItem*.

  The value of `subType` **MUST** be one of the `DataItemSubTypeEnum` enumeration.

- `keyType`

  `type` that defines the meaning of `key`.

  The value of `keyType` **MUST** be one of the `DataItemTypeEnum` enumeration.

6.3.2.2 Part Properties of EntryDefinition

Table 84 lists the Part Properties of `EntryDefinition`.

Part Property name	Multiplicity
CellDefinition (organized by CellDefinitions)	0..*

Table 84: Part Properties of EntryDefinition

Descriptions for Part Properties of EntryDefinition:

- CellDefinition

  semantic definition of a Cell.

  CellDefinitions groups one or more CellDefinition entities if representation is TABLE. See *Section 6.3.1 - CellDefinition*.

6.4 Relationship Types for DataItem

This section provides semantic information for the types of AbstractDataItemRelationship that can be defined for a DataItem.

6.4.1 AbstractDataItemRelationship

association between a DataItem and another entity.

AbstractDataItemRelationship is an abstract entity and hence will be realized by specific AbstractDataItemRelationship types in an MTConnectDevices entity. See *Section 6.4 - Relationship Types for DataItem*.

6.4.1.1 Value Properties of AbstractDataItemRelationship

Table 85 lists the Value Properties of AbstractDataItemRelationship.

Value Property name	Value Property type	Multiplicity
idRef	ID	1
name	string	0..1

Table 85: Value Properties of AbstractDataItemRelationship

Descriptions for Value Properties of AbstractDataItemRelationship:

- idRef
  reference to the related entity's id.
- name
  descriptive name associated with this AbstractDataItemRelationship.

6.4.2 SpecificationRelationship

AbstractDataItemRelationship that provides a semantic reference to another Specification described by type and idRef.

6.4.2.1 Value Properties of SpecificationRelationship

Table 86 lists the Value Properties of SpecificationRelationship.

Value Property name	Value Property type	Multiplicity
type	SpecificationRelationshipTypeEnum	1

Table 86: Value Properties of SpecificationRelationship

Descriptions for Value Properties of SpecificationRelationship:

- type
  specifies how the Specification is related.
  SpecificationRelationshipTypeEnum Enumeration:
  - LIMIT
    referenced Specification provides process limits.

6.4.3 DataItemRelationship

AbstractDataItemRelationship that provides a semantic reference to another DataItem described by type.

6.4.3.1 Value Properties of DataItemRelationship

Table 87 lists the Value Properties of DataItemRelationship.

Value Property name	Value Property type	Multiplicity
type	DataItemRelationshipTypeEnum	1

Table 87: Value Properties of DataItemRelationship

Descriptions for Value Properties of DataItemRelationship:

- type

  specifies how the DataItem is related.

  DataItemRelationshipTypeEnum Enumeration:

  - ATTACHMENT
    reference to a DataItem that associates the values with an external entity.
  - COORDINATE_SYSTEM
    referenced DataItem provides the id of the effective Coordinate System.
  - LIMIT
    referenced DataItem provides process limits.
  - OBSERVATION
    referenced DataItem provides the observed values.

6.5 DataItem Types

This section provides semantic information for the types of a DataItem.

In the MTConnect Standard, DataItem are defined and organized based upon the category and type. The category provides a high level grouping for DataItems based on the kind of information that is reported by the data item.

These categories are:

- SAMPLE: A SAMPLE reports a continuously variable or analog data value.
- EVENT: An EVENT reports information representing a functional state, with two or more discrete values, associated with a component or it contains a message. The data provided may be a numeric value or text.
- CONDITION: A CONDITION reports information about the health of a piece of equipment and its ability to function.

The type specifies the specific kind of data that is reported. For some types of data items, a subType may also be used to differentiate between multiple data items of the same type where the information reported by the data item has a different, but related, meaning.

Many types of data items provide two forms of data: a value (reported as either a SAMPLE or EVENT) and a health status (reported as a CONDITION). These DataItem types **MAY** be defined in more than one category based on the data that they report.

6.5.1 Condition

abstract DataItem that is about an entity's status regarding its ability to operate or it provides an indication whether the data reported for the entity is within an expected range.

Condition is reported differently than Sample or Event. Condition **MUST** be reported as Normal, Warning, or Fault.

All Samples **MAY** have associated Condition states. Condition states indicate whether the value for the data is within an expected range and **MUST** be reported as Normal, or the value is unexpected or out of tolerance for the data and a Warning or Fault **MUST** be provided.

6.5.1.1 Value Properties of Condition

Table 88 lists the Value Properties of Condition.

Value Property name	Value Property type	Multiplicity
category	CONDITION	1
type	ConditionEnum	1

Table 88: Value Properties of Condition

Descriptions for Value Properties of Condition:

- type

  <<extensible>> ConditionEnum Enumeration:

  - ACTUATOR

    indication of a fault associated with an actuator.

- COMMUNICATIONS
  indication that the piece of equipment has experienced a communications failure.
- DATA_RANGE
  indication that the value of the data associated with a measured value or a calculation is outside of an expected range.
- LOGIC_PROGRAM
  indication that an error occurred in the logic program or programmable logic controller (PLC) associated with a piece of equipment.
- MOTION_PROGRAM
  indication that an error occurred in the motion program associated with a piece of equipment.
- SYSTEM
  general purpose indication associated with an electronic component of a piece of equipment or a controller that represents a fault that is not associated with the operator, program, or hardware.

6.5.2 Event

abstract DataItem that is a discrete piece of information from a piece of equipment. It does not have intermediate values that vary over time.

An Event is information that, when provided at any specific point in time, represents the current state of the piece of equipment.

6.5.2.1 Value Properties of Event

Table 89 lists the Value Properties of Event.

Value Property name	Value Property type	Multiplicity
category	EVENT	1
type	EventEnum	1

Table 89: Value Properties of Event

Descriptions for Value Properties of Event:

- type

<<extensible>> EventEnum Enumeration:

- ACTIVATION_COUNT
  accumulation of the number of times a function has attempted to, or is planned to attempt to, activate or be performed.
- ACTIVE_AXES
  set of axes currently associated with a Path or Controller.
- ACTIVE_POWER_SOURCE
  active energy source for the Component.
- ACTUATOR_STATE
  operational state of an apparatus for moving or controlling a mechanism or system.
- ADAPTER_SOFTWARE_VERSION
  originator's software version of the *adapter*.
- ADAPTER_URI
  Uniform Resource Identifier (URI) of the *adapter*.
- <<deprecated>> ALARM
  DEPRECATED: Replaced with CONDITION category data items in Version 1.1.0.
- <<deprecated>> ALARM_LIMIT
  set of limits used to trigger warning or alarm indicators.
  DEPRECATED in *Version 2.5*. Replaced by ALARM_LIMITS.
- ALARM_LIMITS
  set of limits used to trigger warning or alarm indicators.
- APPLICATION
  application on a Component.
- ASSET_CHANGED
  assetId of the *Asset* that has been added or changed.
- ASSET_COUNT
  data set of the number of *Assets* of a given type for a *Device*.
- ASSET_REMOVED
  assetId of the *Asset* that has been removed.
- AVAILABILITY
  agent's ability to communicate with the data source.

- AXIS_COUPLING

  describes the way the axes will be associated to each other.

  This is used in conjunction with COUPLED_AXES to indicate the way they are interacting.

- AXIS_FEEDRATE_OVERRIDE

  value of a signal or calculation issued to adjust the feedrate of an individual linear type axis.

- AXIS_INTERLOCK

  state of the axis lockout function when power has been removed and the axis is allowed to move freely.

- AXIS_STATE

  state of a Linear or Rotary component representing an axis.

- BATTERY_STATE

  present status of the battery.

- BLOCK

  line of code or command being executed by a Controller entity.

- BLOCK_COUNT

  total count of the number of blocks of program code that have been executed since execution started.

- CHARACTERISTIC_PERSISTENT_ID

  UUID of the *characteristic*.

- CHARACTERISTIC_STATUS

  pass/fail result of the measurement.

- CHUCK_INTERLOCK

  state of an interlock function or control logic state intended to prevent the associated Chuck component from being operated.

- CHUCK_STATE

  operating state of a mechanism that holds a part or stock material during a manufacturing process.

  It may also represent a mechanism that holds any other mechanism in place within a piece of equipment.

- CLOCK_TIME

  time provided by a timing device at a specific point in time.

- <<deprecated>> CODE

  programmatic code being executed.

  DEPRECATED in *Version 1.1*.

- COMPONENT_DATA
  Event that represents a Component where the EntryDefinition identifies the Component and the CellDefinitions define the Component's observed DataItems.
- COMPOSITION_STATE
  operating state of a mechanism represented by a Composition entity.
- CONNECTION_STATUS
  status of the connection between an *adapter* and an *agent*.
- CONTROLLER_MODE
  current mode of the Controller component.
- CONTROLLER_MODE_OVERRIDE
  setting or operator selection that changes the behavior of a piece of equipment.
- <<deprecated>> CONTROL_LIMIT
  set of limits used to indicate whether a process variable is stable and in control.
  DEPRECATED in *Version 2.5*. Replaced by CONTROL_LIMITS.
- CONTROL_LIMITS
  set of limits used to indicate whether a process variable is stable and in control.
- COUPLED_AXES
  set of associated axes.
- CYCLE_COUNT
  accumulation of the number of times a cyclic function has attempted to, or is planned to attempt to execute.
- DATE_CODE
  time and date code associated with a material or other physical item.
- DEACTIVATION_COUNT
  accumulation of the number of times a function has attempted to, or is planned to attempt to, deactivate or cease.
- DEVICE_ADDED
  UUID of new device added to an *MTConnect Agent*.
- DEVICE_CHANGED
  UUID of the device whose *metadata* has changed.
- DEVICE_REMOVED
  UUID of a device removed from an *MTConnect Agent*.
- DEVICE_UUID
  identifier of another piece of equipment that is temporarily associated with a component of this piece of equipment to perform a particular function.

- DIRECTION
  direction of motion.
- DOOR_STATE
  operational state of a Door component or composition element.
- EMERGENCY_STOP
  state of the emergency stop signal for a piece of equipment, controller path, or any other component or subsystem of a piece of equipment.
- END_OF_BAR
  indication of whether the end of a piece of bar stock being feed by a bar feeder has been reached.
- EQUIPMENT_MODE
  indication that a piece of equipment, or a sub-part of a piece of equipment, is performing specific types of activities.
- EXECUTION
  operating state of a Component.
- FEATURE_MEASUREMENT
  assessing elements of a *feature*.
- FEATURE_PERSISTENT_ID
  UUID of a *feature*. *Ref ISO 10303 AP 242/239*.
- FIRMWARE
  embedded software of a Component .
- FIXTURE_ID
  identifier for the current workholding or part clamp in use by a piece of equipment.
- FUNCTIONAL_MODE
  current intended production status of the Component.
- HARDNESS
  hardness of a material.
- HARDWARE
  hardware of a Component.
- HOST_NAME
  name of the host computer supplying data.
- LEAK_DETECT
  indication designating whether a leak has been detected.

- LIBRARY
  software library on a Component
- <<deprecated>> LINE
  current line of code being executed.
  DEPRECATED in *Version 1.4.0*.
- LINE_LABEL
  identifier for a Block of code in a Program.
- LINE_NUMBER
  position of a block of program code within a control program.
- LOAD_COUNT
  accumulation of the number of times an operation has attempted to, or is planned to attempt to, load materials, parts, or other items.
- LOCATION_ADDRESS
  structured information that allows the unambiguous determination of an object for purposes of identification and location. *Ref ISO 19160-4:2017*
- LOCATION_NARRATIVE
  textual description of the location of an object or activity.
- LOCATION_SPATIAL_GEOGRAPHIC
  absolute geographic location defined by two coordinates, longitude and latitude and an elevation.
- LOCK_STATE
  state or operating mode of a Lock.
- MAINTENANCE_LIST
  actions or activities to be performed in support of a piece of equipment.
- MATERIAL
  identifier of a material used or consumed in the manufacturing process.
- MATERIAL_LAYER
  identifies the layers of material applied to a part or product as part of an additive manufacturing process.
- MEASUREMENT_TYPE
  class of measurement being performed. *Ref QIF 3:2018 Section 6.3*
- MEASUREMENT_UNITS
  engineering units of the measurement.
- MEASUREMENT_VALUE
  measurement based on the measurement type.

- MESSAGE
  information to be transferred from a piece of equipment to a client software application.
- MTCONNECT_VERSION
  reference version of the MTConnect Standard supported by the *adapter*.
- NETWORK
  network details of a Component.
- NETWORK_PORT
  number of the TCP/IP or UDP/IP port for the connection endpoint.
- OPERATING_MODE
  state of Component or Composition that describes the automatic or manual operation of the entity.
- OPERATING_SYSTEM
  Operating System (OS) of a Component.
- OPERATOR_ID
  identifier of the person currently responsible for operating the piece of equipment.
- PALLET_ID
  identifier for a pallet.
- PART_COUNT
  aggregate count of parts.
- PART_COUNT_TYPE
  interpretation of PART_COUNT.
- PART_DETECT
  indication designating whether a part or work piece has been detected or is present.
- PART_GROUP_ID
  identifier given to a collection of individual parts.
- PART_ID
  identifier of a part in a manufacturing operation.
- PART_INDEX
  sequence of a part in a group of parts.
- PART_KIND_ID
  identifier given to link the individual occurrence to a class of parts, typically distinguished by a particular part design.

- <<deprecated>> PART_NUMBER
  identifier of a part or product moving through the manufacturing process.
  DEPRECATED in *Version 1.7*. PART_NUMBER is now a subType of PART_KIND_ID.
- PART_PROCESSING_STATE
  particular condition of the part occurrence at a specific time.
- PART_STATUS
  state or condition of a part.
- PART_UNIQUE_ID
  identifier given to a distinguishable, individual part.
- PATH_FEEDRATE_OVERRIDE
  value of a signal or calculation issued to adjust the feedrate for the axes associated with a Path component that may represent a single axis or the coordinated movement of multiple axes.
- PATH_MODE
  describes the operational relationship between a Path entity and another Path entity for pieces of equipment comprised of multiple logical groupings of controlled axes or other logical operations.
- POWER_STATE
  indication of the status of the source of energy for an entity to allow it to perform its intended function or the state of an enabling signal providing permission for the entity to perform its functions.
- <<deprecated>> POWER_STATUS
  status of the Component.
  DEPRECATED in *Version 1.1.0*.
- PROCESS_AGGREGATE_ID
  identifier given to link the individual occurrence to a group of related occurrences, such as a process step in a process plan.
- PROCESS_KIND_ID
  identifier given to link the individual occurrence to a class of processes or process definition.
- PROCESS_OCCURRENCE_ID
  identifier of a process being executed by the device.
- PROCESS_STATE
  particular condition of the process occurrence at a specific time.
- PROCESS_TIME
  time and date associated with an activity or event.

- PROGRAM
  name of the logic or motion program being executed by the Controller component.
- PROGRAM_COMMENT
  comment or non-executable statement in the control program.
- PROGRAM_EDIT
  indication of the status of the Controller components program editing mode.
  A program may be edited while another is executed.
- PROGRAM_EDIT_NAME
  name of the program being edited.
  This is used in conjunction with ProgramEdit when in ACTIVE state.
- PROGRAM_HEADER
  non-executable header section of the control program.
- PROGRAM_LOCATION
  URI for the source file associated with Program.
- PROGRAM_LOCATION_TYPE
  defines whether the logic or motion program defined by Program is being executed from the local memory of the controller or from an outside source.
- PROGRAM_NEST_LEVEL
  indication of the nesting level within a control program that is associated with the code or instructions that is currently being executed.
- ROTARY_MODE
  current operating mode for a Rotary type axis.
- ROTARY_VELOCITY_OVERRIDE
  percentage change to the velocity of the programmed velocity for a Rotary axis.
- ROTATION
  three space angular displacement of an object or coordinate system relative to a *cartesian coordinate system*.
- SENSOR_ATTACHMENT
  attachment between a sensor and an entity.
- SENSOR_STATE
  detection result of a sensor.
- SERIAL_NUMBER
  serial number associated with a Component, Asset, or Device.

- <<deprecated>> SPECIFICATION_LIMIT

  set of limits defining a range of values designating acceptable performance for a variable.

  DEPRECATED in *Version 2.5*. Replaced by SPECIFICATION_LIMITS.

- SPECIFICATION_LIMITS

  set of limits defining a range of values designating acceptable performance for a variable.

- SPINDLE_INTERLOCK

  indication of the status of the spindle for a piece of equipment when power has been removed and it is free to rotate.

- THICKNESS

  dimension between two surfaces of an object, usually the dimension of smallest measure, for example an additive layer, or a depth of cut.

- TOOL_ASSET_ID

  identifier of an individual tool asset.

- TOOL_CUTTING_ITEM

  references the CuttingToolLifeCycle CuttingItem index related to the indices of the currently active cutting tool edge.

- TOOL_GROUP

  identifier for the tool group associated with a specific tool. Commonly used to designate spare tools.

- <<deprecated>> TOOL_ID

  identifier of the tool currently in use for a given Path.

  DEPRECATED in *Version 1.2.0*. See TOOL_NUMBER.

- TOOL_NUMBER

  identifier assigned by the Controller component to a cutting tool when in use by a piece of equipment.

- TOOL_OFFSET

  reference to the tool offset variables applied to the active cutting tool.

- TOOL_OFFSETS

  properties of each addressable tool offset.

- TRANSFER_COUNT

  accumulation of the number of times an operation has attempted to, or is planned to attempt to, transfer materials, parts, or other items from one location to another.

- TRANSLATION
  three space linear displacement of an object or coordinate system relative to a *cartesian coordinate system*.
- UNCERTAINTY
  uncertainty specified by UncertaintyType.
- UNCERTAINTY_TYPE
  method used to compute *standard uncertainty*.
- UNLOAD_COUNT
  accumulation of the number of times an operation has attempted to, or is planned to attempt to, unload materials, parts, or other items.
- USER
  identifier of the person currently responsible for operating the piece of equipment.
- VALVE_STATE
  state of a valve is one of open, closed, or transitioning between the states.
- VARIABLE
  data whose meaning may change over time due to changes in the operation of a piece of equipment or the process being executed on that piece of equipment.
- WAIT_STATE
  indication of the reason that Execution is reporting a value of WAIT.
- WIRE
  identifier for the type of wire used as the cutting mechanism in Electrical Discharge Machining or similar processes.
- WORKHOLDING_ID
  identifier for the current workholding or part clamp in use by a piece of equipment.
  DEPRECATION WARNING: Recommend using FIXTURE_ID instead.
- WORK_OFFSET
  reference to offset variables for a work piece or part.
- WORK_OFFSETS
  properties of each addressable work offset.

6.5.3 Sample

abstract DataItem that is continuously changing or analog data value.

This data can be measured at any point-in-time and will always produce a result.

The `units` for `Sample` **MUST** always be specified.

6.5.3.1 Value Properties of Sample

Table 90 lists the Value Properties of `Sample`.

Value Property name	Value Property type	Multiplicity
`category`	`SAMPLE`	1
`type`	`SampleEnum`	1

Table 90: Value Properties of Sample

Descriptions for Value Properties of `Sample`:

- `type`
  `<<extensible>>` `SampleEnum` Enumeration:
  - `ACCELERATION`
    positive rate of change of velocity.
  - `ACCUMULATED_TIME`
    accumulated time for an activity or event.
  - `<<deprecated>>` `AMPERAGE`
    strength of electrical current.
    DEPRECATED in *Version 1.6*. Replaced by `AMPERAGE_AC` and `AMPERAGE_DC`.
  - `AMPERAGE_AC`
    electrical current that reverses direction at regular short intervals.
  - `AMPERAGE_DC`
    electric current flowing in one direction only.
  - `ANGLE`
    angular position.
  - `ANGULAR_ACCELERATION`
    positive rate of change of angular velocity.
  - `ANGULAR_DECELERATION`
    negative rate of change of angular velocity.

- ANGULAR_VELOCITY
  rate of change of angular position.
- ASSET_UPDATE_RATE
  average rate of change of values for assets in the MTConnect streams.
  The average is computed over a rolling window defined by the implementation.
- AXIS_FEEDRATE
  feedrate of a linear axis.
- BATTERY_CAPACITY
  maximum rated charge a battery is capable of maintaining based on the battery discharging at a specified current over a specified time period.
- BATTERY_CHARGE
  value of the battery's present capacity expressed as a percentage of the battery's maximum rated capacity.
- CAPACITY_FLUID
  maximum amount of fluid that can be held by a container.
- CAPACITY_SPATIAL
  maximum amount of material that can be held by a container.
- CHARGE_RATE
  value of the current being supplied to the Component for the purpose of charging.
- CONCENTRATION
  percentage of one component within a mixture of components.
- CONDUCTIVITY
  ability of a material to conduct electricity.
- CUTTING_SPEED
  speed difference (relative velocity) between the cutting mechanism and the surface of the workpiece it is operating on.
- DECELERATION
  negative rate of change of velocity.
- DENSITY
  volumetric mass of a material per unit volume of that material.
- DEPOSITION_ACCELERATION_VOLUMETRIC
  rate of change in spatial volume of material deposited in an additive manufacturing process.

- DEPOSITION_DENSITY
  density of the material deposited in an additive manufacturing process per unit of volume.
- DEPOSITION_MASS
  mass of the material deposited in an additive manufacturing process.
- DEPOSITION_RATE_VOLUMETRIC
  rate at which a spatial volume of material is deposited in an additive manufacturing process.
- DEPOSITION_VOLUME
  spatial volume of material to be deposited in an additive manufacturing process.
- DEW_POINT
  temperature at which moisture begins to condense, corresponding to saturation for a given absolute humidity.
- DIAMETER
  dimension of a diameter.
- DISCHARGE_RATE
  value of current being drawn from the Component.
- DISPLACEMENT
  change in position of an object.
- DISPLACEMENT_ANGULAR
  absolute value of the change in angular position around a vector
- DISPLACEMENT_LINEAR
  absolute value of the change in position along a vector.
- ELECTRICAL_ENERGY
  Wattage used or generated by a component over an interval of time.
- EQUIPMENT_TIMER
  amount of time a piece of equipment or a sub-part of a piece of equipment has performed specific activities.
- FILL_HEIGHT
  amount of a substance in a container.
- FILL_LEVEL
  amount of a substance remaining compared to the planned maximum amount of that substance.
- FLOW
  rate of flow of a fluid.

- FOLLOWING_ERROR
  difference between actual and commanded position at any specific point in time during a motion.
- FOLLOWING_ERROR_ANGULAR
  angular difference between the commanded encoder/resolver position and the actual encoder/resolver position at any specified point in time during a motion.
- FOLLOWING_ERROR_LINEAR
  difference between the commanded encoder/resolver position and the actual encoder/resolver position at any specified point in time during a motion.
- FREQUENCY
  number of occurrences of a repeating event per unit time.
- <<deprecated>> GLOBAL_POSITION
  position in three-dimensional space.
  DEPRECATED in Version 1.1.
- GRAVITATIONAL_ACCELERATION
  acceleration relative to Earth's gravity of 9.80665 METER/SECOND^2.
- GRAVITATIONAL_FORCE
  force relative to earth's gravity.
- HUMIDITY_ABSOLUTE
  amount of water vapor expressed in grams per cubic meter.
- HUMIDITY_RELATIVE
  amount of water vapor present expressed as a percent to reach saturation at the same temperature.
- HUMIDITY_SPECIFIC
  ratio of the water vapor present over the total weight of the water vapor and air present expressed as a percent.
- LENGTH
  length of an object.
- <<deprecated>> LEVEL
  level of a resource.
  DEPRECATED in *Version 1.2*. See FILL_LEVEL.
- LINEAR_FORCE
  force applied to a mass in one direction only.
- LOAD
  actual versus the standard rating of a piece of equipment.

- MASS
  mass of an object(s) or an amount of material.
- OBSERVATION_UPDATE_RATE
  average rate of change of values for data items in the MTConnect streams. The average is computed over a rolling window defined by the implementation.
- OPENNESS
  percentage open where 100% is fully open and 0% is fully closed.
- ORIENTATION
  angular position of a plane or vector relative to a *cartesian coordinate system*
- PARTICLE_COUNT
  number of particles counted by their size or other characteristics.
- PARTICLE_SIZE
  size of particles counted by their size or other characteristics.
- PATH_FEEDRATE
  feedrate for the axes, or a single axis, associated with a `Path` component.
- PATH_FEEDRATE_PER_REVOLUTION
  feedrate for the axes, or a single axis.
- PATH_POSITION
  position of a control point associated with a `Controller` or a `Path`.
- PH
  acidity or alkalinity of a solution.
- POSITION
  point along an axis in a *cartesian coordinate system*.
- POSITION_CARTESIAN
  point in a *cartesian coordinate system*.
- POWER_FACTOR
  ratio of real power flowing to a load to the apparent power in that AC circuit.
- PRESSURE
  force per unit area measured relative to atmospheric pressure.
  Commonly referred to as gauge pressure.
- PRESSURE_ABSOLUTE
  force per unit area measured relative to a vacuum.
- PRESSURIZATION_RATE
  change of pressure per unit time.

- PROCESS_TIMER
  amount of time a piece of equipment has performed different types of activities associated with the process being performed at that piece of equipment.
- RESISTANCE
  degree to which a substance opposes the passage of an electric current.
- RESISTIVITY
  inability of a material to conduct electricity.
- ROTARY_VELOCITY
  rotational speed of a rotary axis.
- SETTLING_ERROR
  difference between actual and commanded position at the end of a motion.
- SETTLING_ERROR_ANGULAR
  angular difference between the commanded encoder/resolver position, and the actual encoder/resolver position when motion is complete.
- SETTLING_ERROR_LINEAR
  difference between the commanded encoder/resolver position, and the actual encoder/resolver position when motion is complete.
- SOUND_LEVEL
  sound level or sound pressure level relative to atmospheric pressure.
- <<deprecated>> SPINDLE_SPEED
  rotational speed of the rotary axis.
  DEPRECATED in *Version 1.2*. Replaced by ROTARY_VELOCITY.
- STRAIN
  amount of deformation per unit length of an object when a load is applied.
- TEMPERATURE
  degree of hotness or coldness measured on a definite scale.
- TENSION
  force that stretches or elongates an object.
- TILT
  angular displacement.
- TORQUE
  turning force exerted on an object or by an object.
- VELOCITY
  rate of change of position of a Component.

- VISCOSITY
  fluid's resistance to flow.
- <<deprecated>> VOLTAGE
  electrical potential between two points.
  DEPRECATED in *Version 1.6*. Replaced by VOLTAGE_AC and VOLTAGE_DC.
- VOLTAGE_AC
  electrical potential between two points in an electrical circuit in which the current periodically reverses direction.
- VOLTAGE_DC
  electrical potential between two points in an electrical circuit in which the current is unidirectional.
- VOLT_AMPERE
  apparent power in an electrical circuit, equal to the product of root-mean-square (RMS) voltage and RMS current (commonly referred to as VA).
- VOLT_AMPERE_REACTIVE
  reactive power in an AC electrical circuit (commonly referred to as VAR).
- VOLUME_FLUID
  fluid volume of an object or container.
- VOLUME_SPATIAL
  geometric volume of an object or container.
- WATTAGE
  power flowing through or dissipated by an electrical circuit or piece of equipment.
- X_DIMENSION
  dimension of an entity relative to the X direction of the referenced coordinate system.
- Y_DIMENSION
  dimension of an entity relative to the Y direction of the referenced coordinate system.
- Z_DIMENSION
  dimension of an entity relative to the Z direction of the referenced coordinate system.

7 References Model

References *organizes* pointers to information defined elsewhere within the MTConnectDevices entity for a piece of equipment. It is an efficient method to associate information with an element without duplicating any of the data or structure.

7.1 References

This section provides semantic information for the Reference entity. Figure 6 shows the Reference model.

Reference may be modeled as part of a Device, Component or Interface type.

> Note: See *Section B.3 - References Schema Diagrams* for XML schema of Reference and its types.

7.1.1 Reference

pointer to information that is associated with another entity defined elsewhere in the MTConnectDevices entity for a piece of equipment.

Reference is an abstract entity and will be realized by a specific Reference type for an MTConnectDevices entity. See *Section 7.1.3 - ComponentRef* and *Section 7.1.2 - DataItemRef*.

7.1.1.1 Value Properties of Reference

Table 91 lists the Value Properties of Reference.

Value Property name	Value Property type	Multiplicity
idRef	ID	1
name	string	0..1

Table 91: Value Properties of Reference

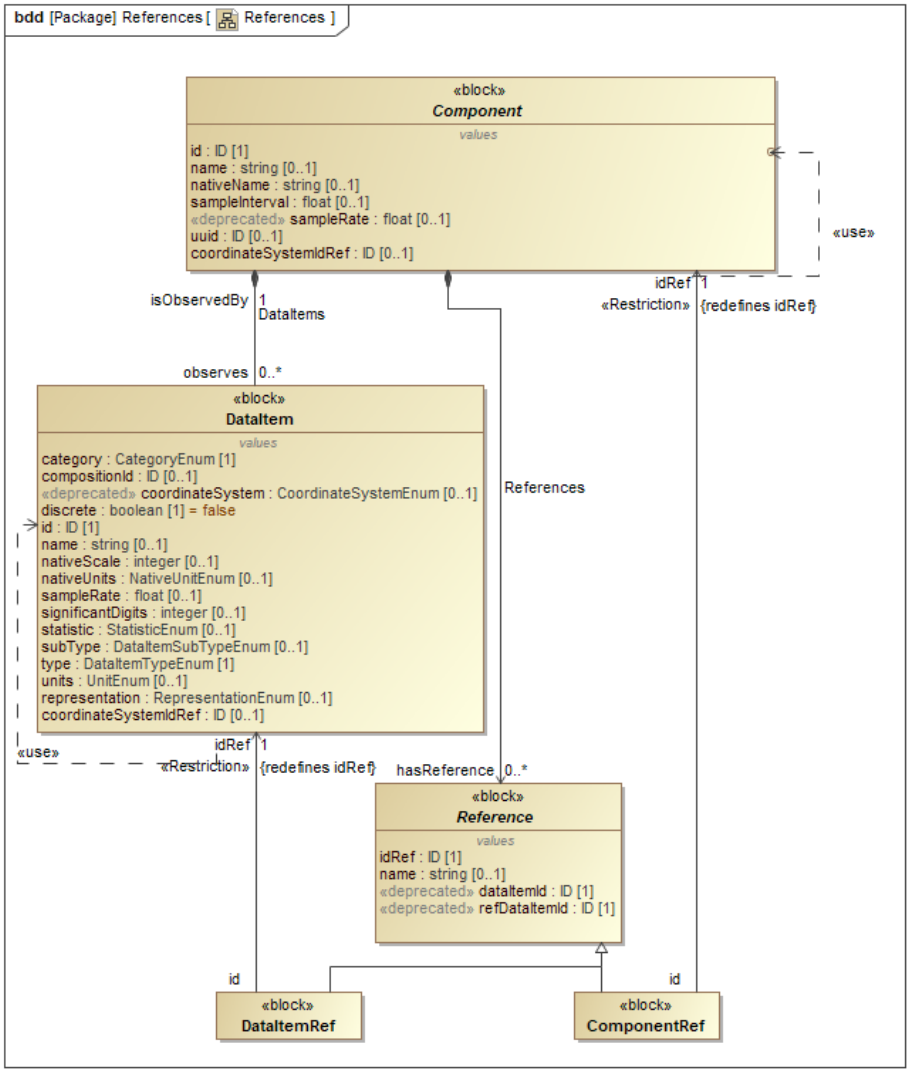

Figure 6: References

Descriptions for Value Properties of `Reference`:

- `idRef`

  pointer to the `id` of an entity that contains the information to be associated with this entity.

- `name`

  name of an element or a piece of equipment.

7.1.2 DataItemRef

`Reference` that is a pointer to a `DataItem` associated with another entity defined for a piece of equipment.

`DataItemRef` allows the data associated with a `DataItem` defined in another entity to be directly associated with this entity.

7.1.3 ComponentRef

`Reference` that is a pointer to all of the information associated with another entity defined for a piece of equipment.

`ComponentRef` allows all of the information of (*lower level* `Component` entities that is associated with the other entity to be directly associated with this entity.

3569 8 Configurations Model

3570 This section provides semantic information for the `Configuration` entity that is used
3571 to model technical information about a `Component`.

3572 8.1 Configurations

3573 Figure 7 shows the abstract `Configuration` and its types.

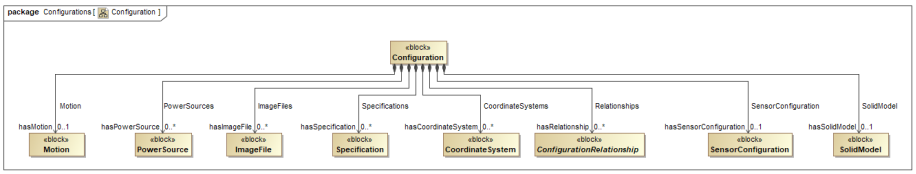

Figure 7: Configuration

3574 Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

3575 8.1.1 Configuration

3576 technical information about an entity describing its physical layout, functional character-
3577 istics, and relationships with other entities.

3578 8.1.1.1 Part Properties of Configuration

3579 *Table 92* lists the Part Properties of `Configuration`.

Part Property name	Multiplicity
SolidModel	0..1
SensorConfiguration	0..1
Motion	0..1
ConfigurationRelationship (organized by Relationships)	0..*
CoordinateSystem (organized by CoordinateSystems)	0..*
Specification (organized by Specifications)	0..*
ImageFile (organized by ImageFiles)	0..*
PowerSource (organized by PowerSources)	0..*

Table 92: Part Properties of Configuration

Descriptions for Part Properties of Configuration:

- SolidModel

  references to a file with the three-dimensional geometry of the Component or Composition.

  See *Section 8.6 - SolidModel*.

- SensorConfiguration

  configuration for a Sensor.

  See SensorConfiguration.

- Motion

  movement of the component relative to a coordinate system.

  See *Section 8.3 - Motion*.

- ConfigurationRelationship

  association between two pieces of equipment that function independently but together perform a manufacturing operation.

  Relationships groups one or more ConfigurationRelationship types. See *Section 8.4 - Relationships*.

- CoordinateSystem

  reference system that associates a unique set of n parameters with each point in an n-dimensional space. *Ref ISO 10303-218:2004*

  CoordinateSystems groups one or more CoordinateSystem entities. See *Section 8.2 - CoordinateSystems*.

- `Specification`

  design characteristics for a piece of equipment.

  `Specifications` groups one or more `Specification` entities. See *Section 8.7 - Specifications*.

- `ImageFile`

  reference to a file containing an image of the `Component`.

  `ImageFiles` groups one or more `ImageFile` entities. See *Section 8.8 - ImageFiles*.

- `PowerSource`

  potential energy sources for the `Component`.

  `PowerSources` groups one or more `PowerSource` entities. See *Section 8.9 - PowerSources*.

8.2 CoordinateSystems

This section provides semantic information for the `CoordinateSystem` entity.

Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

Note: See *Example 6* for the XML representation of the same example.

8.2.1 CoordinateSystem

reference system that associates a unique set of n parameters with each point in an n-dimensional space. *Ref ISO 10303-218:2004*

At most only one of `Origin` or `Transformation` **MUST** be defined for a `CoordinateSystem`.

8.2.1.1 Value Properties of CoordinateSystem

Table 93 lists the Value Properties of `CoordinateSystem`.

Value Property name	Value Property type	Multiplicity
id	ID	1
name	string	0..1
nativeName	string	0..1
parentIdRef	ID	0..1
type	CoordinateSystemTypeEnum	1
uuid	UUID	0..1
Description	string	0..1

Table 93: Value Properties of CoordinateSystem

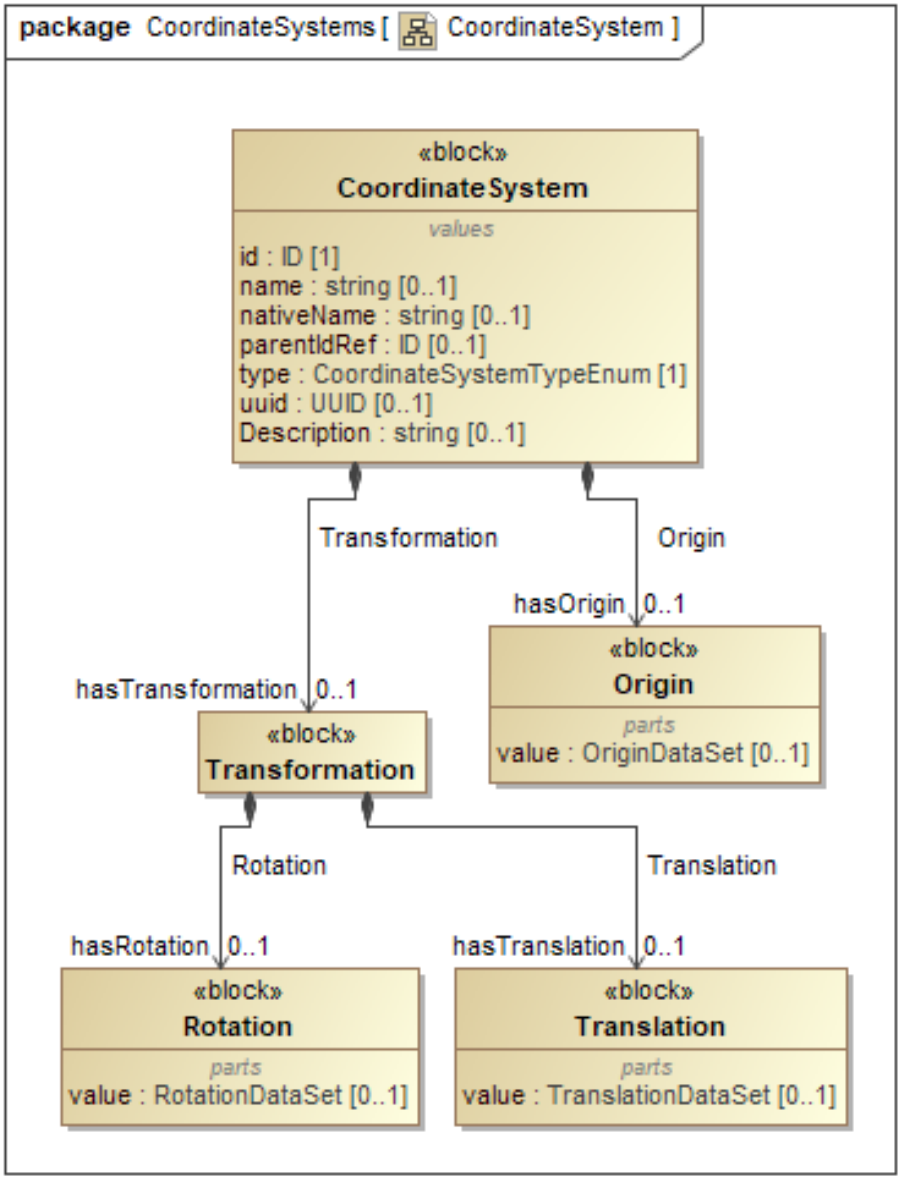

Figure 8: CoordinateSystem

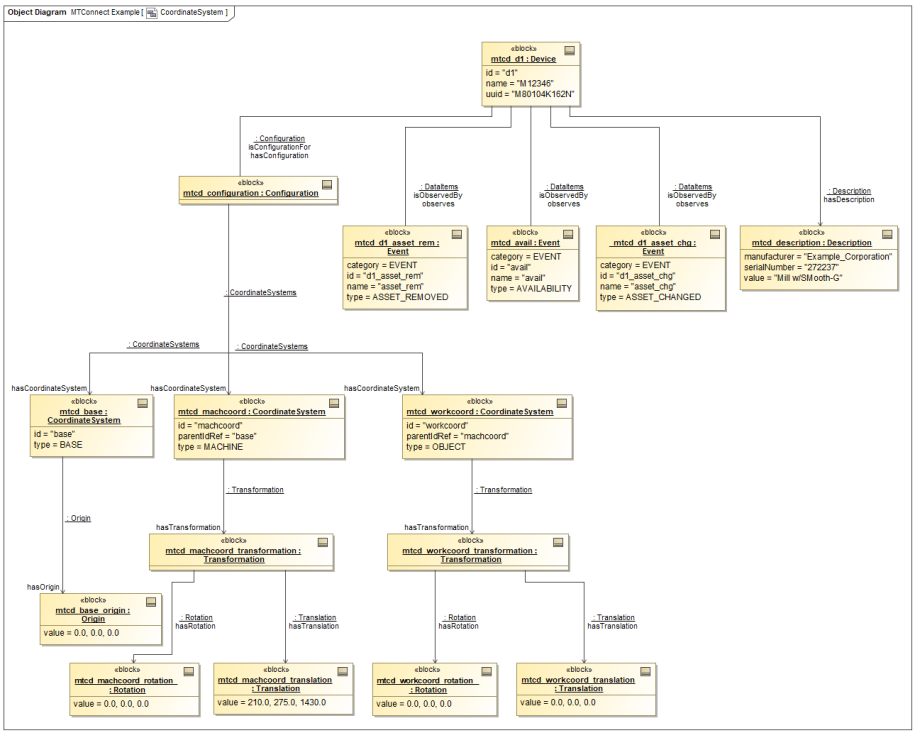

Figure 9: CoordinateSystem Example

Descriptions for Value Properties of `CoordinateSystem`:

- `id`
  unique identifier for the coordinate system.
- `name`
  name of the coordinate system.
- `nativeName`
  manufacturer's name or users name for the coordinate system.
- `parentIdRef`
  pointer to the `id`.
- `type`
  type of coordinate system.
  `CoordinateSystemTypeEnum` Enumeration:
  - `BASE`
    coordinate system referenced to the base mounting surface. *Ref ISO 9787:2013*
    A base mounting surface is a connection surface between the arm and its supporting structure.*Ref ISO 9787:2013*
    For non-robotic devices, it is the connection surface between the device and its supporting structure.
  - `CAMERA`
    coordinate system referenced to the sensor which monitors the site of the task. *Ref ISO 9787:2013*
  - `MACHINE`
    coordinate system referenced to the home position and orientation of the primary axes of a piece of equipment.
  - `MECHANICAL_INTERFACE`
    coordinate system referenced to the mechanical interface. *Ref ISO 9787:2013*
  - `MOBILE_PLATFORM`
    coordinate system referenced to one of the components of a mobile platform. *Ref ISO 8373:2012*
  - `OBJECT`
    coordinate system referenced to the object. *Ref ISO 9787:2013*

- TASK

  coordinate system referenced to the site of the task. *Ref ISO 9787:2013*

- TOOL

  coordinate system referenced to the tool or to the end effector attached to the mechanical interface. *Ref ISO 9787:2013*

- WORLD

  stationary coordinate system referenced to earth, which is independent of the robot motion. *Ref ISO 9787:2013*

  For non-robotic devices, stationary coordinate system referenced to earth, which is independent of the motion of a piece of equipment.

- uuid

  UUID for the coordinate system.

- Description

  natural language description of the CoordinateSystem.

8.2.1.2 Part Properties of CoordinateSystem

Table 94 lists the Part Properties of CoordinateSystem.

Part Property name	Multiplicity
Origin	0..1
Transformation	0..1

Table 94: Part Properties of CoordinateSystem

Descriptions for Part Properties of CoordinateSystem:

- Origin

  coordinates of the origin position of a coordinate system.

  See *Section 8.2.2 - Origin*.

- Transformation

  process of transforming to the origin position of the coordinate system from a parent coordinate system using Translation and Rotation.

  See *Section 8.2.3 - Transformation*.

8.2.2 Origin

coordinates of the origin position of a coordinate system.

8.2.2.1 Part Properties of Origin

Table 95 lists the Part Properties of `Origin`.

Part Property name	Multiplicity
`value`	0..1

Table 95: Part Properties of Origin

Descriptions for Part Properties of `Origin`:

- `OriginDataSet`

  x-coordinate.

8.2.3 Transformation

process of transforming to the origin position of the coordinate system from a parent coordinate system using `Translation` and `Rotation`.

At a minimum, a `Translation` or a `Rotation` **MUST** be defined for a `Transformation`.

8.2.3.1 Part Properties of Transformation

Table 96 lists the Part Properties of `Transformation`.

Part Property name	Multiplicity
`Translation`	0..1
`Rotation`	0..1

Table 96: Part Properties of Transformation

Descriptions for Part Properties of `Transformation`:

- `Translation`

  translations along X, Y, and Z axes are expressed as x,y, and z respectively within a 3-dimensional vector.

  See *Section 8.2.5 - Translation*.

- `Rotation`

  rotations about X, Y, and Z axes are expressed in A, B, and C respectively within a 3-dimensional vector.

  See *Section 8.2.4 - Rotation*.

8.2.4 Rotation

rotations about X, Y, and Z axes are expressed in A, B, and C respectively within a 3-dimensional vector.

8.2.4.1 Part Properties of Rotation

Table 97 lists the Part Properties of `Rotation`.

Part Property name	Multiplicity
`value`	0..1

Table 97: Part Properties of Rotation

Descriptions for Part Properties of `Rotation`:

- `RotationDataSet`

  rotation about X axis.

8.2.5 Translation

translations along X, Y, and Z axes are expressed as x,y, and z respectively within a 3-dimensional vector.

8.2.5.1 Part Properties of Translation

Table 98 lists the Part Properties of `Translation`.

Part Property name	Multiplicity
value	0..1

Table 98: Part Properties of Translation

Descriptions for Part Properties of Translation:

- TranslationDataSet
  translation along X axis.

8.3 Motion

This section provides semantic information for the Motion entity.

Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

Note: See *Example 7* for the XML representation of the same example.

8.3.1 Motion

movement of the component relative to a coordinate system.

Motion specifies the kinematic chain of the component entities.

At most only one of Origin or Transformation **MUST** be defined for a Motion.

8.3.1.1 Value Properties of Motion

Table 99 lists the Value Properties of Motion.

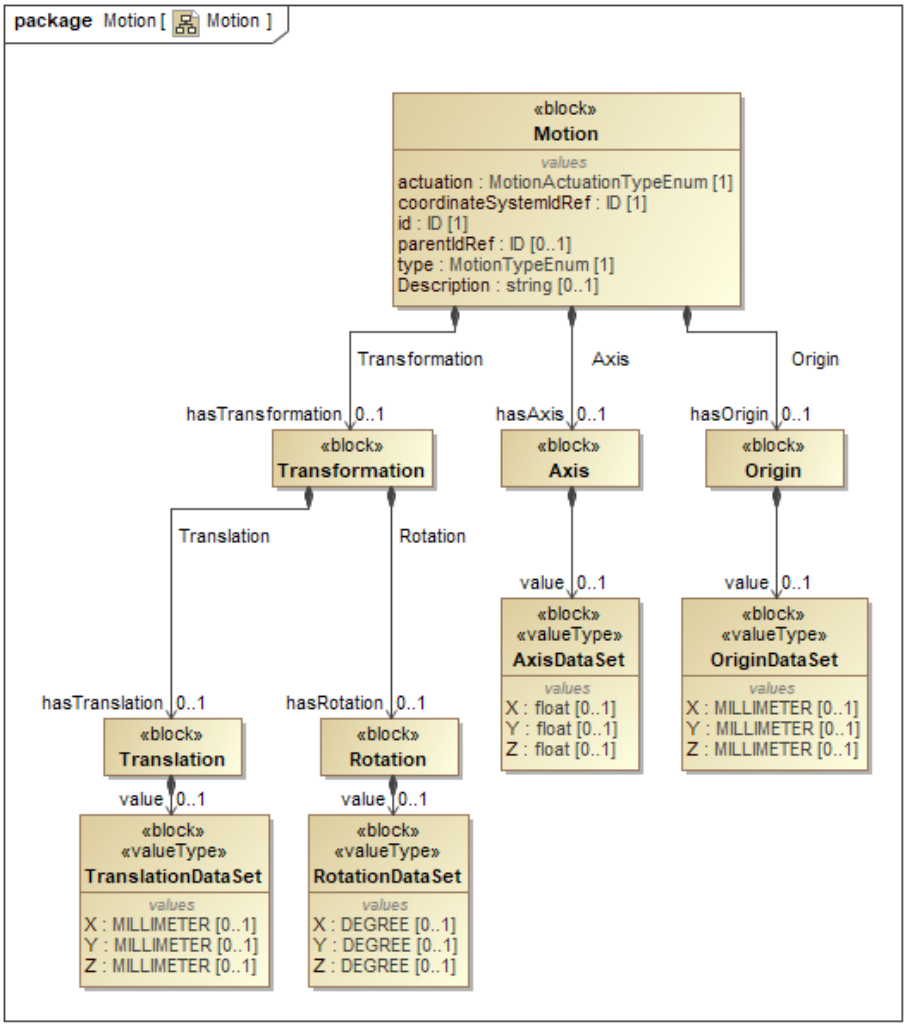

Figure 10: Motion

Value Property name	Value Property type	Multiplicity
actuation	MotionActuationTypeEnum	1
coordinateSystemIdRef	ID	1
id	ID	1
parentIdRef	ID	0..1
type	MotionTypeEnum	1

Table 99: Value Properties of Motion

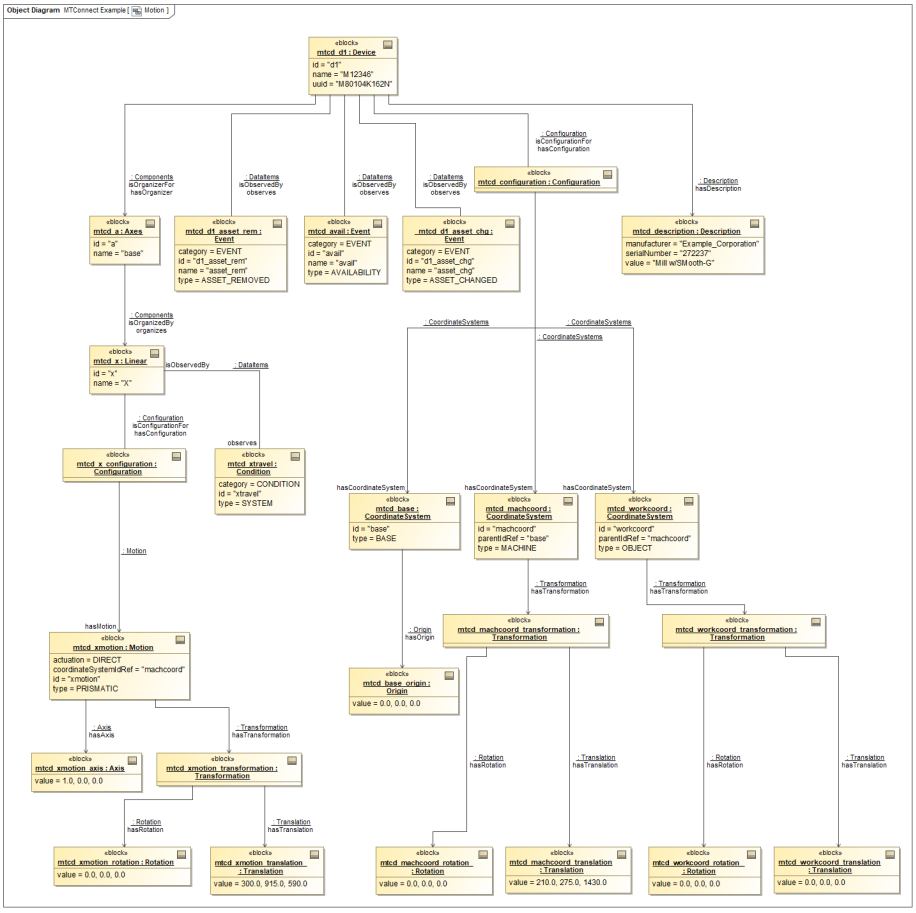

Figure 11: Motion Example

Descriptions for Value Properties of `Motion`:

- `actuation`

  describes if this component is actuated directly or indirectly as a result of other motion.

  `MotionActuationTypeEnum` Enumeration:

  - `DIRECT`
    movement is initiated by the component.
  - `NONE`
    no actuation of this axis.

    Note: Actuation of `NONE` can be either a derived `REVOLUTE` or `PRISMATIC` motion or static `FIXED` relationship.
  - `VIRTUAL`
    motion is computed and is used for expressing an imaginary movement.

- `coordinateSystemIdRef`

  coordinate system within which the kinematic motion occurs.

- `id`

  unique identifier for this element.

- `parentIdRef`

  pointer to the `id`.

  The kinematic chain connects all components using the parent relations. All motion is connected to the motion of the parent. The first node in the chain will not have a parent.

- `type`

  type of motion.

  `MotionTypeEnum` Enumeration:

  - `CONTINUOUS`
    revolves around an axis with a continuous range of motion.
  - `FIXED`
    axis does not move.
  - `PRISMATIC`
    sliding linear motion along an axis with a fixed range of motion.
  - `REVOLUTE`
    rotates around an axis with a fixed range of motion.

8.3.1.2 Part Properties of Motion

Table 100 lists the Part Properties of `Motion`.

Part Property name	Multiplicity
`Axis`	0..1
`Origin`	0..1
`Transformation`	0..1

Table 100: Part Properties of Motion

Descriptions for Part Properties of `Motion`:

- `Axis`
  axis along or around which the `Component` moves relative to a coordinate system.
  See *Section 4.3.4 - Axis*.
- `Origin`
  coordinates of the origin position of a coordinate system.
  See *Section 8.2.2 - Origin*.
- `Transformation`
  process of transforming to the origin position of the coordinate system from a parent coordinate system using `Translation` and `Rotation`.
  See *Section 8.2.3 - Transformation*.

8.3.2 Axis

axis along or around which the `Component` moves relative to a coordinate system.

8.3.2.1 Part Properties of Axis

Table 101 lists the Part Properties of `Axis`.

Part Property name	Multiplicity
`value`	0..1

Table 101: Part Properties of Axis

Descriptions for Part Properties of `Axis`:

- `AxisDataSet`
  x-component of `Axis`.

8.4 Relationships

This section provides semantic information for the `ConfigurationRelationship` entity.

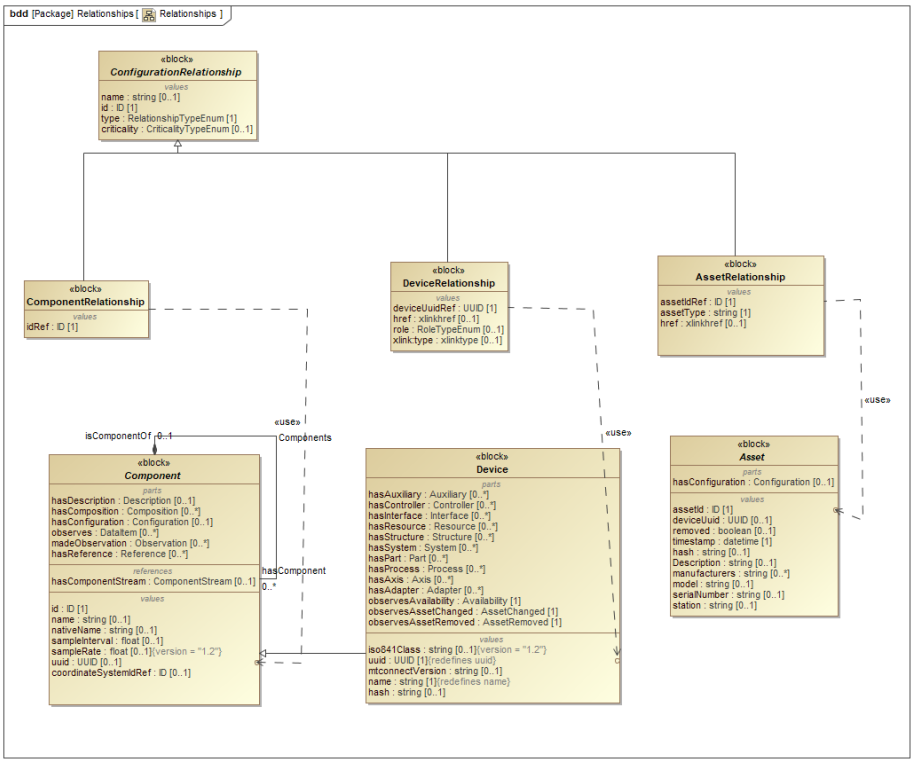

Figure 12: Relationships

Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

Note: See *Example 8* for the XML representation of the same example.

8.4.1 ConfigurationRelationship

association between two pieces of equipment that function independently but together perform a manufacturing operation.

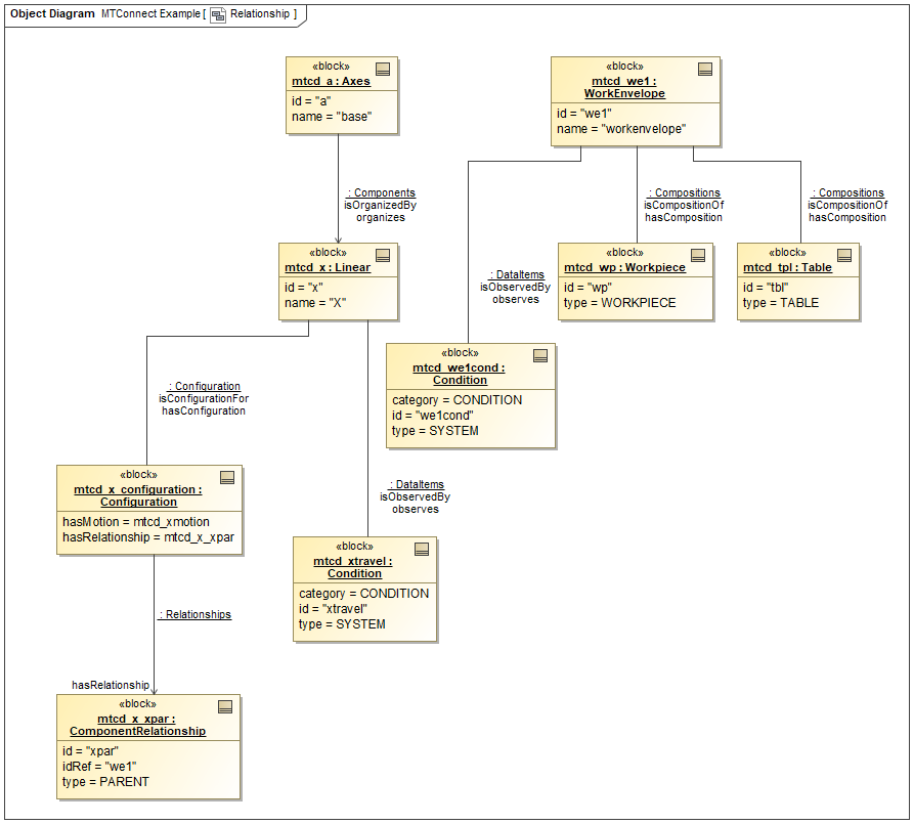

Figure 13: Relationship Example

`ConfigurationRelationship` is an abstract entity and hence will be realized by specific `ConfigurationRelationship` types in an `MTConnectDevices` entity. See *Section 8.4.2 - ComponentRelationship* and *Section 8.4.3 - DeviceRelationship*.

8.4.1.1 Value Properties of ConfigurationRelationship

Table 102 lists the Value Properties of `ConfigurationRelationship`.

Value Property name	Value Property type	Multiplicity
`name`	`string`	0..1
`id`	`ID`	1
`type`	`RelationshipTypeEnum`	1
`criticality`	`CriticalityTypeEnum`	0..1

Table 102: Value Properties of ConfigurationRelationship

Descriptions for Value Properties of `ConfigurationRelationship`:

- `name`

  name associated with this `ConfigurationRelationship`.

- `id`

  unique identifier for this `ConfigurationRelationship`.

- `type`

  defines the authority that this piece of equipment has relative to the associated piece of equipment.

  `RelationshipTypeEnum` Enumeration:

  - `CHILD`

    functions as a child in the relationship with the associated element.

  - `PARENT`

    functions as a parent in the relationship with the associated element.

  - `PEER`

    functions as a peer which provides equal functionality and capabilities in the relationship with the associated element.

- `criticality`

  defines whether the services or functions provided by the associated piece of equipment is required for the operation of this piece of equipment.

  `CriticalityTypeEnum` Enumeration:

  - `CRITICAL`

    services or functions provided by the associated element is required for the operation of this element.

  - `NONCRITICAL`

    services or functions provided by the associated element is not required for the operation of this element.

8.4.2 ComponentRelationship

`ConfigurationRelationship` that describes the association between two components within a piece of equipment that function independently but together perform a capability or service within a piece of equipment.

8.4.2.1 Value Properties of ComponentRelationship

Table 103 lists the Value Properties of `ComponentRelationship`.

Value Property name	Value Property type	Multiplicity
idRef	ID	1

Table 103: Value Properties of ComponentRelationship

Descriptions for Value Properties of `ComponentRelationship`:

- `idRef`

  reference to the associated `Component`.

8.4.3 DeviceRelationship

`ConfigurationRelationship` that describes the association between two pieces of equipment that function independently but together perform a manufacturing operation.

8.4.3.1 Value Properties of DeviceRelationship

Table 104 lists the Value Properties of `DeviceRelationship`.

Value Property name	Value Property type	Multiplicity
deviceUuidRef	UUID	1
href	xlinkhref	0..1
role	RoleTypeEnum	0..1
xlink:type	xlinktype	0..1

Table 104: Value Properties of DeviceRelationship

Descriptions for Value Properties of `DeviceRelationship`:

- `deviceUuidRef`

  reference to the `uuid` of the associated piece of equipment.

- `href`

  URI identifying the *agent* that is publishing information for the associated piece of equipment.

- `role`

  defines the services or capabilities that the referenced piece of equipment provides relative to this piece of equipment.

  `RoleTypeEnum` Enumeration:

  - `AUXILIARY`
    associated element performs the functions as an `Auxiliary` for this element.
  - `SYSTEM`
    associated element performs the functions of a `System` for this element.

- `xlink:type`

  `xlink:type`**MUST** have a fixed value of `locator` as defined in W3C XLink 1.1 *Ref https://www.w3.org/TR/xlink11/.*

8.4.4 AssetRelationship

`ConfigurationRelationship` that describes the association between a `Component` and an `Asset`.

8.4.4.1 Value Properties of AssetRelationship

Table 105 lists the Value Properties of `AssetRelationship`.

Value Property name	Value Property type	Multiplicity
`assetIdRef`	`ID`	1
`assetType`	`string`	1
`href`	`xlinkhref`	0..1

Table 105: Value Properties of AssetRelationship

Descriptions for Value Properties of `AssetRelationship`:

- `assetIdRef`

  uuid of the related `Asset`.

- `assetType`

  type of `Asset` being referenced.

- `href`

  URI reference to the associated `Asset`.

8.5 Sensor

This section provides semantic information for the `SensorConfiguration` entity.

sensor is a unique type of a piece of equipment. A sensor is typically comprised of two major components: a sensor unit that provides signal processing, conversion, and communications and the *sensing elements* that provides a signal or measured value.

The sensor unit is modeled as a *lower level* `Component` called `Sensor`. The *sensing element* may be modeled as a `Composition` element of a `Sensor` element and the measured value would be modeled as a `DataItem` (See *Section 6.1 - DataItems* for more information on `DataItem` elements). Each sensor unit may have multiple *sensing elements*; each representing the data for a variety of measured values.

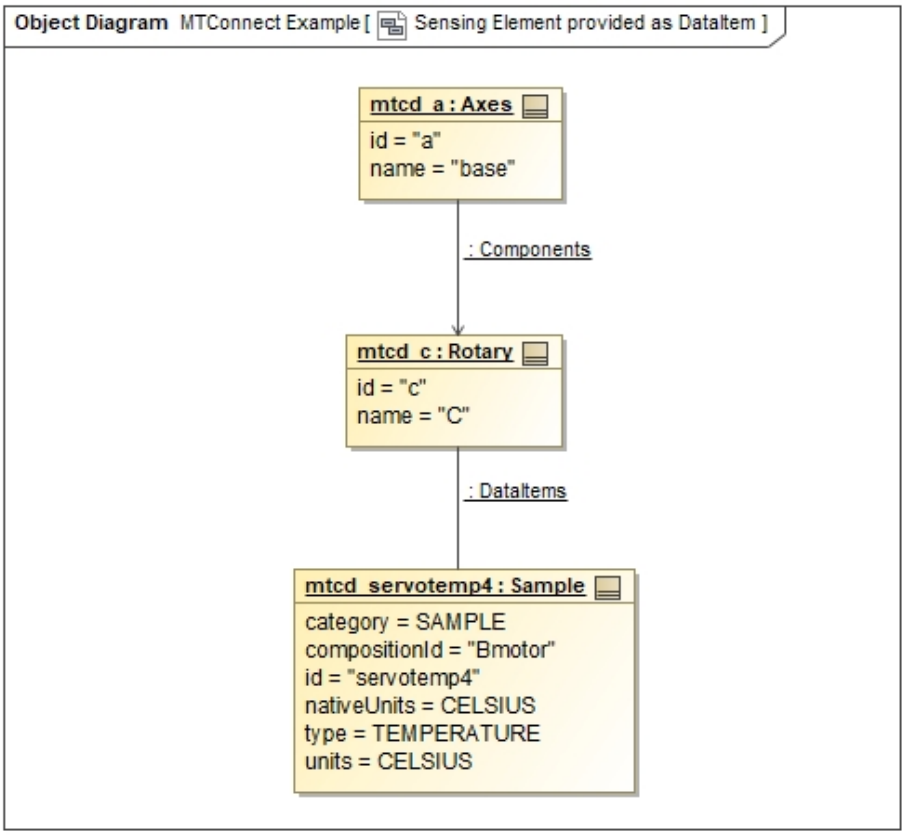

Figure 14: Sensing Element provided as a DataItem Example

Note: See *Example 11* for an XML example.

When a sensor unit is modeled as a `Component` or as a separate piece of equipment, it

may provide additional configuration information for the sensor elements and the sensor unit itself.

Note: If a `Sensor` provides vibration measurement data for the spindle on a piece of equipment, it could be modeled as a `Sensor` for rotary axis named `C`. See *Example 12* for an XML example.

Note: If a `Sensor` provides measurement data for multiple `Component` elements within a piece of equipment and is not associated with any particular `Component`, it **MAY** be modeled as an independent `Component` and the data associated with measurements are associated with their associated `Component` entities. See *Example 13* for an XML example.

`Configuration` data provides information required for maintenance and support of the sensor.

When `Sensor` represents the sensor unit for multiple *sensing element*(s), each sensing element is represented by a `Channel`. The sensor unit itself and each `Channel` representing one *sensing element* **MAY** have its own configuration data.

Note: See *Example 14* for an XML example.

8.5.1 SensorConfiguration

configuration for a `Sensor`.

Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

8.5.1.1 Value Properties of SensorConfiguration

Table 106 lists the Value Properties of `SensorConfiguration`.

Value Property name	Value Property type	Multiplicity
CalibrationDate	datetime	0..1
CalibrationInitials	string	0..1
FirmwareVersion	string	1
NextCalibrationDate	datetime	0..1

Table 106: Value Properties of SensorConfiguration

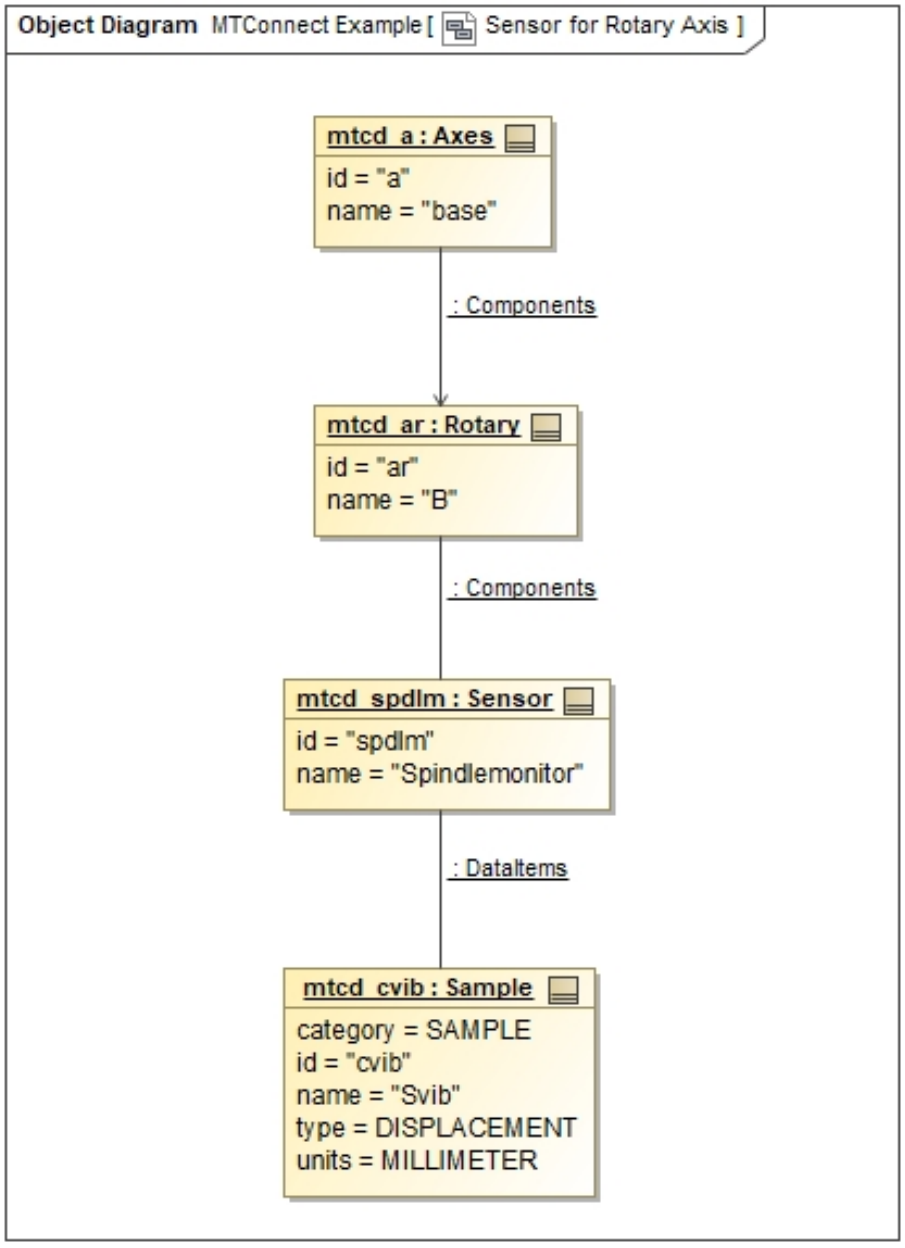

Figure 15: Sensor for Rotary Axis Example

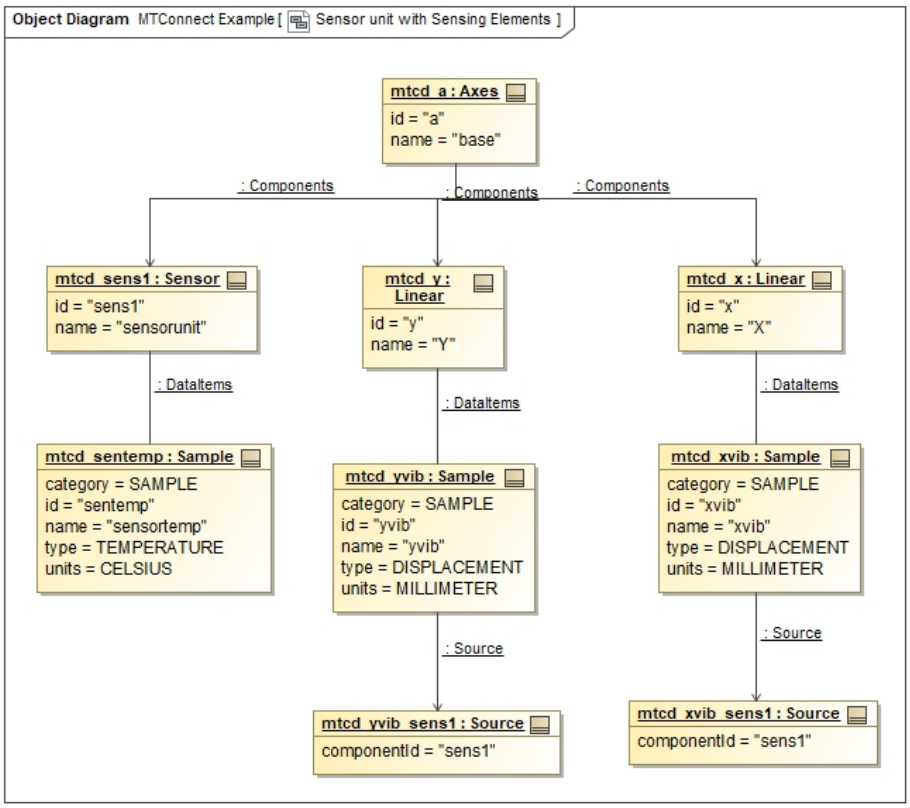

Figure 16: Sensor unit with Sensing Elements Example

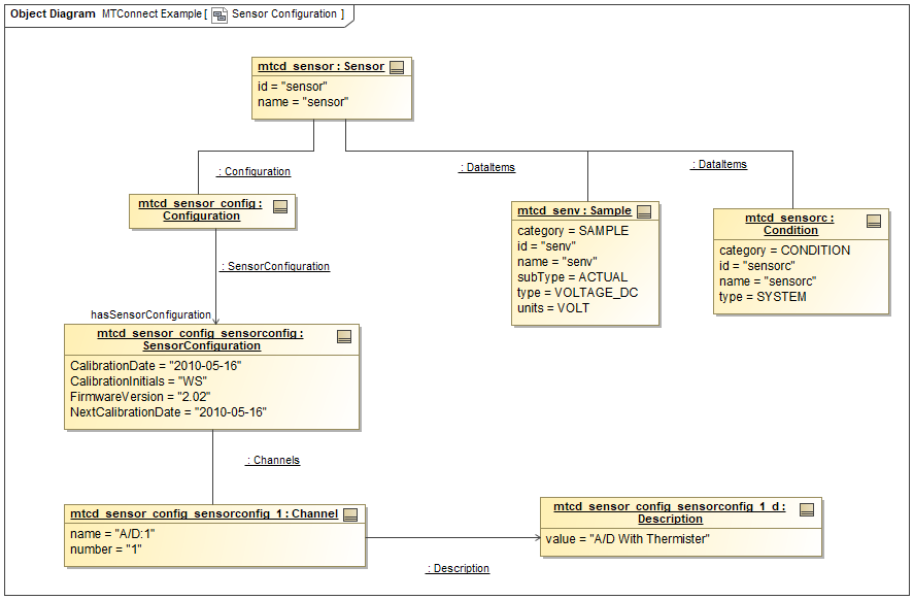

Figure 17: Sensor Configuration Example

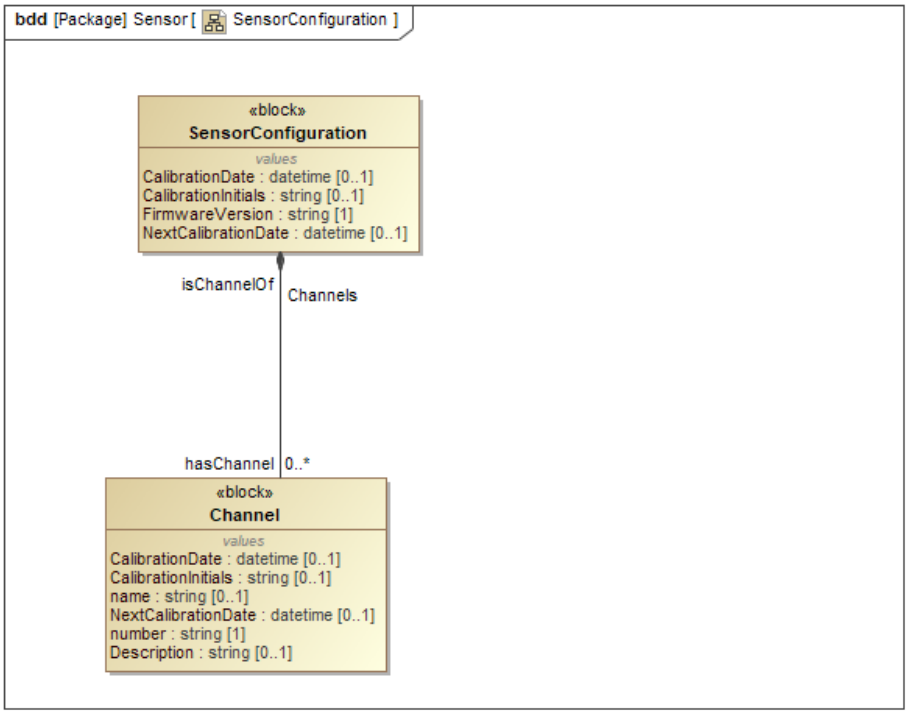

Figure 18: SensorConfiguration

Descriptions for Value Properties of SensorConfiguration:

- CalibrationDate
  Date upon which the sensor unit was last calibrated.
- CalibrationInitials
  The initials of the person verifying the validity of the calibration data.
- FirmwareVersion
  Version number for the sensor unit as specified by the manufacturer.
- NextCalibrationDate
  Date upon which the sensor unit is next scheduled to be calibrated.

8.5.1.2 Part Properties of SensorConfiguration

Table 107 lists the Part Properties of SensorConfiguration.

Part Property name	Multiplicity
Channel (organized by Channels)	0..*

Table 107: Part Properties of SensorConfiguration

Descriptions for Part Properties of SensorConfiguration:

- Channel
  sensing element of a Sensor.
  Channels groups one or more Channel entities. See Channel.

8.5.2 Channel

sensing element of a Sensor.

When Sensor has multiple *sensing elements*, each *sensing element* is modeled as a Channel for the Sensor.

8.5.2.1 Value Properties of Channel

Table 108 lists the Value Properties of Channel.

Value Property name	Value Property type	Multiplicity
CalibrationDate	datetime	0..1
CalibrationInitials	string	0..1
name	string	0..1
NextCalibrationDate	datetime	0..1
number	string	1

Table 108: Value Properties of Channel

Descriptions for Value Properties of `Channel`:

- `CalibrationDate`

  Date upon which the sensor unit was last calibrated to the sensor element.

- `CalibrationInitials`

  The initials of the person verifying the validity of the calibration data.

- `name`

  name of the specific *sensing element*.

- `NextCalibrationDate`

  Date upon which the sensor element is next scheduled to be calibrated with the sensor unit.

- `number`

  unique identifier that will only refer to a specific *sensing element*.

8.6 SolidModel

This section provides semantic information for the `SolidModel` entity.

Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

Note: See *Example 9* for the XML representation of the same example.

8.6.1 SolidModel

references to a file with the three-dimensional geometry of the Component or Composition.

The geometry **MAY** have a transformation and a scale to position the Component with respect to the other Components. A geometry file can contain a set of assembled items, in this case, the SolidModel references the id of the assembly model file and the specific item within that file.

The SolidModel **MAY** provide a translation, rotation, and scale to correctly place it relative to the other geometries in the machine. If the Component can move and has a Motion Configuration, the SolidModel will move when the Component or Composition moves.

Either an href or a modelIdRef and an itemRef **MUST** be specified.

8.6.1.1 Value Properties of SolidModel

Table 109 lists the Value Properties of SolidModel.

Value Property name	Value Property type	Multiplicity
id	ID	1
solidModelIdRef	ID	0..1
href	string	0..1
itemRef	string	0..1
mediaType	MediaTypeEnum	1
coordinateSystemIdRef	ID	0..1
nativeUnits	NativeUnitEnum	0..1
units	UnitEnum	0..1

Table 109: Value Properties of SolidModel

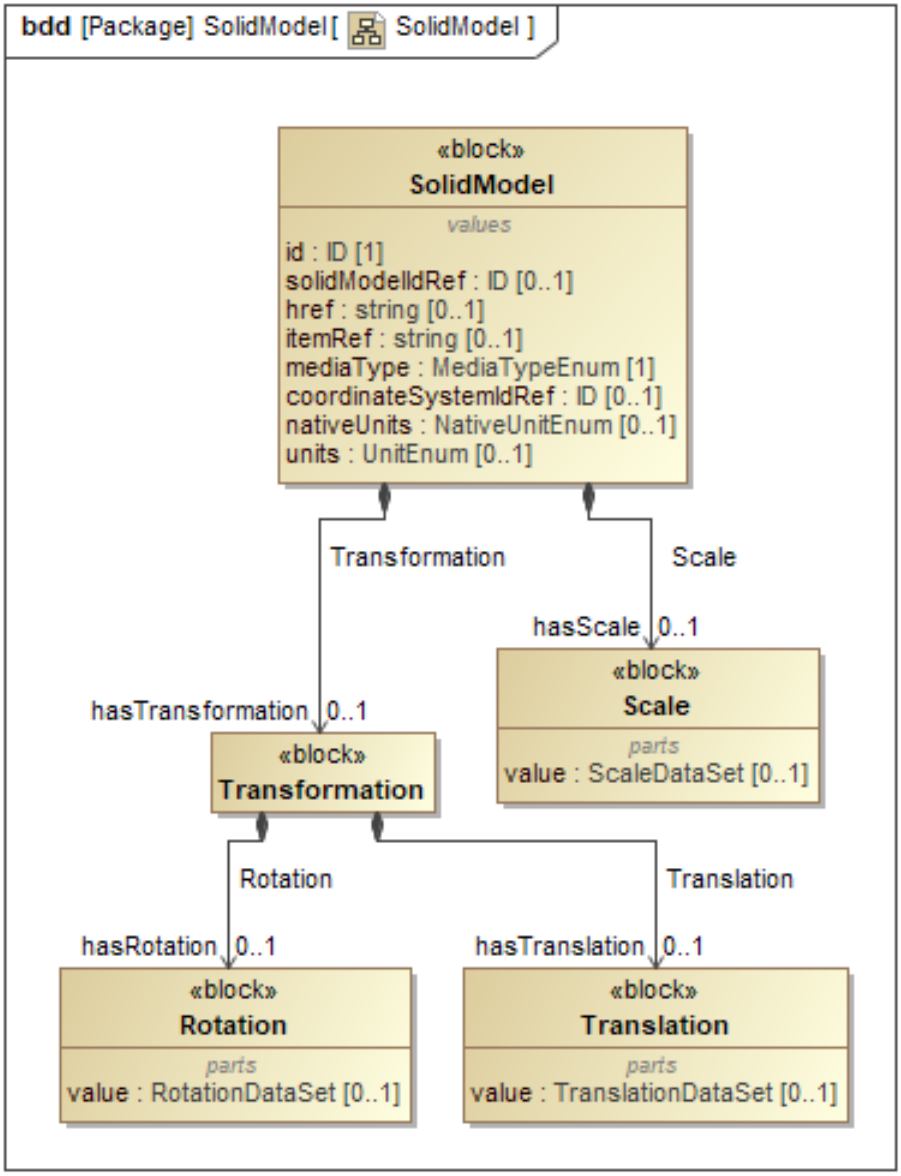

Figure 19: SolidModel

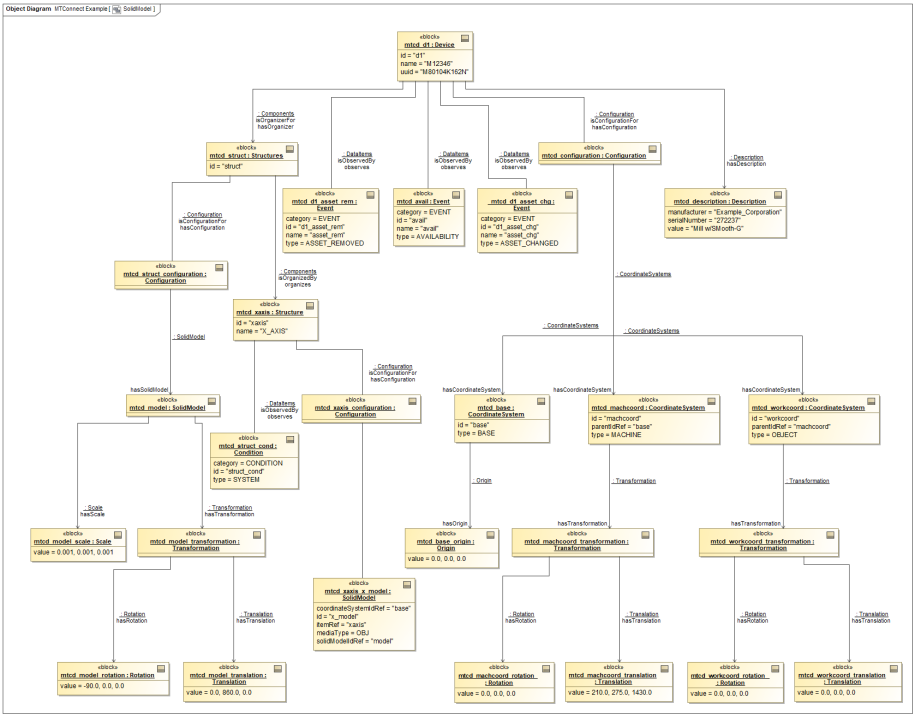

Figure 20: SolidModel Example

Descriptions for Value Properties of SolidModel:

- id

  unique identifier for this element.

- solidModelIdRef

  associated model file if an item reference is used.

- href

  Uniform Resource Locator (URL) giving the location of the SolidModel.

  If not present, the model referenced in the solidModelIdRef is used.

  href is of type xlink:href from the W3C XLink specification.

- itemRef

  reference to the item within the model within the related geometry. A solidModelIdRef **MUST** be given.

  > Note: Item defined in ASME Y14.100 - A nonspecific term used to denote any unit or product, including materials, parts, assemblies, equipment, accessories, and computer software.

- mediaType

  format of the referenced document.

  MediaTypeEnum Enumeration:

  - 3DS

    Autodesk file format.

  - ACIS

    Dassault file format.

  - COLLADA

    ISO 17506.

  - GDML

    Geometry Description Markup Language.

  - IGES

    Initial Graphics Exchange Specification.

  - OBJ

    Wavefront OBJ file format.

- `STEP`
  ISO 10303 STEP AP203 or AP242 format.
- `STL`
  STereoLithography file format.
- `X_T`
  Parasolid XT Siemens data interchange format.

- `coordinateSystemIdRef`

  reference to the coordinate system for this `SolidModel`.

- `nativeUnits`

  same as `nativeUnits`. See *Section 6.1.1 - DataItem*.

  The value of `nativeUnits` **MUST** be one of the `NativeUnitEnum` enumeration.

- `units`

  same as `units`. See *Section 6.1.1 - DataItem*.

  The value of `units` **MUST** be one of the `UnitEnum` enumeration.

8.6.1.2 Part Properties of SolidModel

Table 110 lists the Part Properties of `SolidModel`.

Part Property name	Multiplicity
`Transformation`	0..1
`Scale`	0..1

Table 110: Part Properties of SolidModel

Descriptions for Part Properties of `SolidModel`:

- `Transformation`

  process of transforming to the origin position of the coordinate system from a parent coordinate system using `Translation` and `Rotation`.

  See *Section 8.2.3 - Transformation*.

- `Scale`

  either a single multiplier applied to all three dimensions or a three space multiplier given in the X, Y, and Z dimensions in the coordinate system used for the `SolidModel`.

  See *Section 8.6.2 - Scale*.

8.6.2 Scale

either a single multiplier applied to all three dimensions or a three space multiplier given in the X, Y, and Z dimensions in the coordinate system used for the `SolidModel`.

8.6.2.1 Part Properties of Scale

Table 111 lists the Part Properties of `Scale`.

Part Property name	Multiplicity
`value`	0..1

Table 111: Part Properties of Scale

Descriptions for Part Properties of `Scale`:

- `ScaleDataSet`

  multiplier for X axis.

8.7 Specifications

This section provides semantic information for the `Specification` entity.

Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

Note: See *Example 10* for the XML representation of the same example.

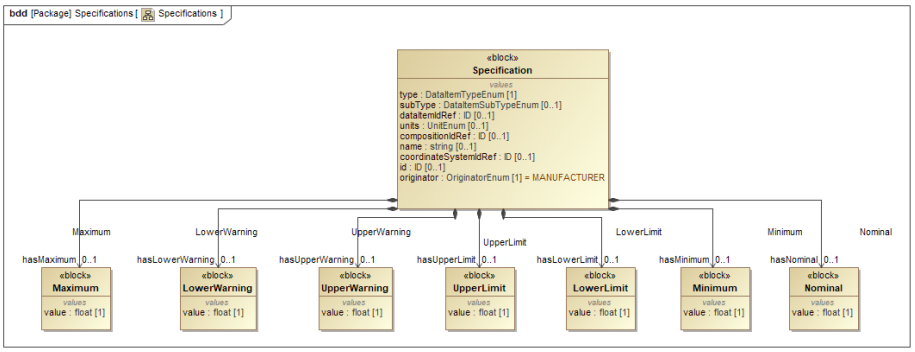

Figure 21: Specifications

8.7.1 Specification

design characteristics for a piece of equipment.

8.7.1.1 Value Properties of Specification

Table 112 lists the Value Properties of `Specification`.

Value Property name	Value Property type	Multiplicity
`type`	`DataItemTypeEnum`	1
`subType`	`DataItemSubTypeEnum`	0..1
`dataItemIdRef`	`ID`	0..1
`units`	`UnitEnum`	0..1
`compositionIdRef`	`ID`	0..1
`name`	`string`	0..1
`coordinateSystemIdRef`	`ID`	0..1
`id`	`ID`	0..1
`originator`	`OriginatorEnum`	1

Table 112: Value Properties of Specification

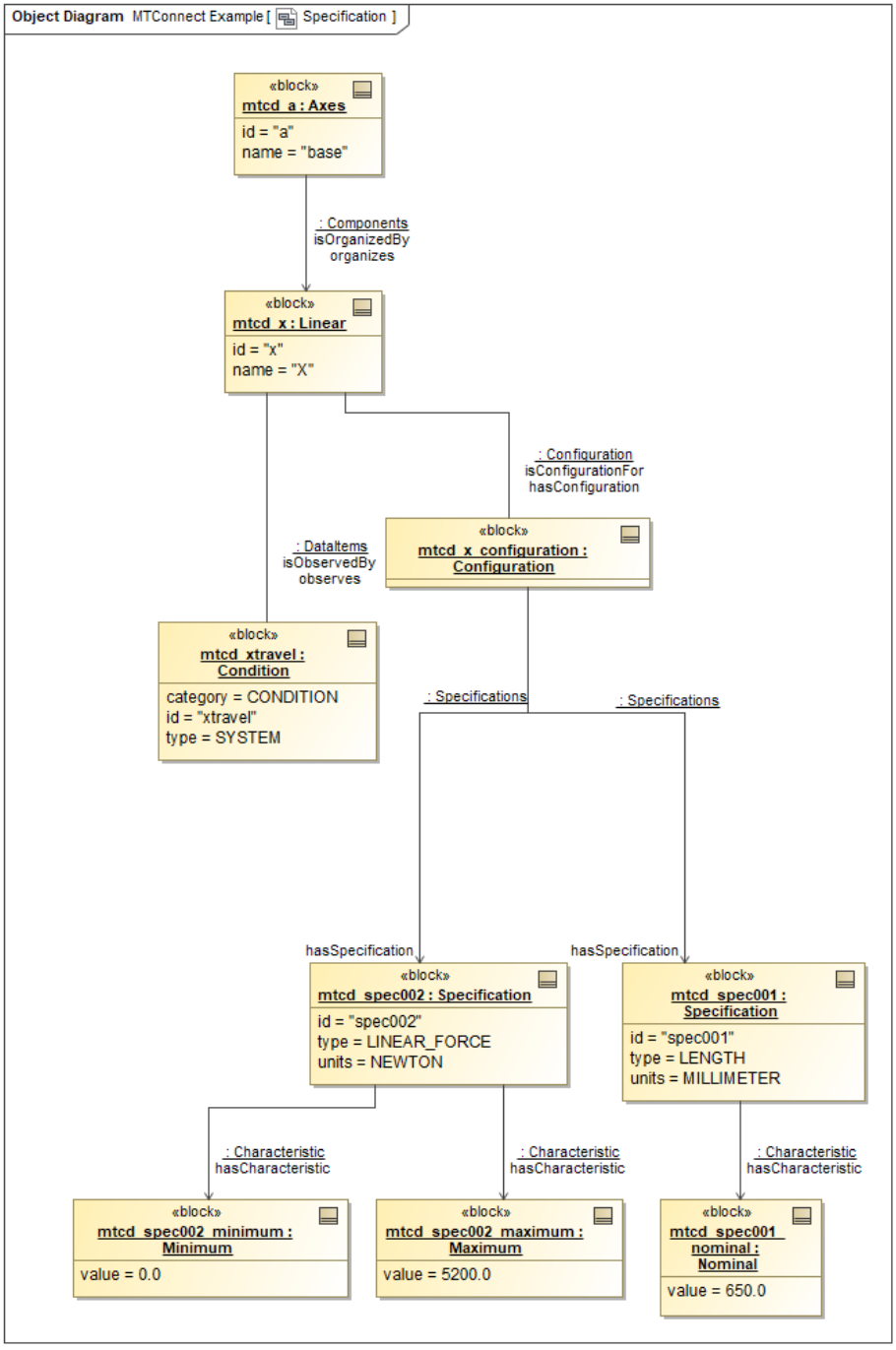

Figure 22: Specification Example

Descriptions for Value Properties of `Specification`:

- `type`

  same as `type`. See *Section 6.5 - DataItem Types*.

  The value of `type` **MUST** be one of the `DataItemTypeEnum` enumeration.

- `subType`

  same as `subType`. See *Section 6.1.1 - DataItem*.

  The value of `subType` **MUST** be one of the `DataItemSubTypeEnum` enumeration.

- `dataItemIdRef`

  reference to the `id` associated with this entity.

- `units`

  same as `units`. See *Section 6.1.1 - DataItem*.

  The value of `units` **MUST** be one of the `UnitEnum` enumeration.

- `compositionIdRef`

  reference to the `id` associated with this entity.

- `name`

  `name` provides additional meaning and differentiates between `Specification` entities.

- `coordinateSystemIdRef`

  references the `CoordinateSystem` for geometric `Specification` elements.

- `id`

  unique identifier for this `Specification`.

- `originator`

  reference to the creator of the `Specification`.

  `OriginatorEnum` Enumeration:

  - `MANUFACTURER`
    manufacturer of a piece of equipment or `Component`.
  - `USER`
    owner or implementer of a piece of equipment or `Component`.

8.7.1.2 Part Properties of Specification

Table 113 lists the Part Properties of `Specification`.

Part Property name	Multiplicity
`Maximum`	0..1
`UpperLimit`	0..1
`LowerWarning`	0..1
`LowerLimit`	0..1
`UpperWarning`	0..1
`Nominal`	0..1
`Minimum`	0..1

Table 113: Part Properties of Specification

Descriptions for Part Properties of `Specification`:

- `Maximum`
  numeric upper constraint.
- `UpperLimit`
  upper conformance boundary for a variable.

  Note: immediate concern or action may be required.
- `LowerWarning`
  lower boundary indicating increased concern and supervision may be required.
- `LowerLimit`
  lower conformance boundary for a variable.

  Note: immediate concern or action may be required.
- `UpperWarning`
  upper boundary indicating increased concern and supervision may be required.
- `Nominal`
  numeric target or expected value.
- `Minimum`
  numeric lower constraint.

8.7.2 ProcessSpecification

Specification that provides information used to assess the conformance of a variable to process requirements.

8.7.2.1 Part Properties of ProcessSpecification

Table 114 lists the Part Properties of ProcessSpecification.

Part Property name	Multiplicity
SpecificationLimits	0..1
ControlLimits	0..1
AlarmLimits	0..1

Table 114: Part Properties of ProcessSpecification

Descriptions for Part Properties of ProcessSpecification:

- SpecificationLimits

  set of limits that define a range of values designating acceptable performance for a variable.

  See *Section 8.7.5 - SpecificationLimits*.

- ControlLimits

  set of limits that is used to indicate whether a process variable is stable and in control.

  See *Section 8.7.3 - ControlLimits*.

- AlarmLimits

  set of limits that is used to trigger warning or alarm indicators.

  See *Section 8.7.4 - AlarmLimits*.

8.7.3 ControlLimits

set of limits that is used to indicate whether a process variable is stable and in control.

8.7.3.1 Part Properties of ControlLimits

Table 115 lists the Part Properties of ControlLimits.

Part Property name	Multiplicity
UpperLimit	0..1
UpperWarning	0..1
LowerWarning	0..1
Nominal	0..1
LowerLimit	0..1

Table 115: Part Properties of ControlLimits

Descriptions for Part Properties of `ControlLimits`:

- `UpperLimit`
  upper conformance boundary for a variable.

  Note: immediate concern or action may be required.

- `UpperWarning`
  upper boundary indicating increased concern and supervision may be required.

- `LowerWarning`
  lower boundary indicating increased concern and supervision may be required.

- `Nominal`
  numeric target or expected value.

- `LowerLimit`
  lower conformance boundary for a variable.

  Note: immediate concern or action may be required.

8.7.4 AlarmLimits

set of limits that is used to trigger warning or alarm indicators.

8.7.4.1 Part Properties of AlarmLimits

Table 116 lists the Part Properties of `AlarmLimits`.

Part Property name	Multiplicity
UpperLimit	0..1
UpperWarning	0..1
LowerLimit	0..1
LowerWarning	0..1

Table 116: Part Properties of AlarmLimits

Descriptions for Part Properties of AlarmLimits:

- UpperLimit
  upper conformance boundary for a variable.

  Note: immediate concern or action may be required.

- UpperWarning
  upper boundary indicating increased concern and supervision may be required.

- LowerLimit
  lower conformance boundary for a variable.

  Note: immediate concern or action may be required.

- LowerWarning
  lower boundary indicating increased concern and supervision may be required.

8.7.5 SpecificationLimits

set of limits that define a range of values designating acceptable performance for a variable.

8.7.5.1 Part Properties of SpecificationLimits

Table 117 lists the Part Properties of SpecificationLimits.

Part Property name	Multiplicity
UpperLimit	0..1
Nominal	0..1
LowerLimit	0..1

Table 117: Part Properties of SpecificationLimits

Descriptions for Part Properties of SpecificationLimits:

- UpperLimit
  upper conformance boundary for a variable.

  Note: immediate concern or action may be required.

- Nominal
  numeric target or expected value.

- LowerLimit
  lower conformance boundary for a variable.

  Note: immediate concern or action may be required.

8.7.6 UpperWarning

upper boundary indicating increased concern and supervision may be required.

The value of UpperWarning **MUST** be float.

8.7.7 UpperLimit

upper conformance boundary for a variable.

Note: immediate concern or action may be required.

The value of UpperLimit **MUST** be float.

8.7.8 Maximum

numeric upper constraint.

The value of `Maximum` **MUST** be `float`.

8.7.9 LowerLimit

lower conformance boundary for a variable.

Note: immediate concern or action may be required.

The value of `LowerLimit` **MUST** be `float`.

8.7.10 LowerWarning

lower boundary indicating increased concern and supervision may be required.

The value of `LowerWarning` **MUST** be `float`.

8.7.11 Minimum

numeric lower constraint.

The value of `Minimum` **MUST** be `float`.

8.7.12 Nominal

numeric target or expected value.

The value of `Nominal` **MUST** be `float`.

8.8 ImageFiles

This section provides semantic information for the ImageFile entity.

8.8.1 ImageFile

reference to a file containing an image of the Component.

8.8.1.1 Value Properties of ImageFile

Table 118 lists the Value Properties of ImageFile.

Value Property name	Value Property type	Multiplicity
id	ID	1
href	string	1
mediaType	string	1
name	string	0..1

Table 118: Value Properties of ImageFile

Descriptions for Value Properties of ImageFile:

- id
  unique identifier of the image file.
- href
  URL giving the location of the image file.
- mediaType
  mime type of the image file.
- name
  description of the image file.

8.9 PowerSources

This section provides semantic information for the PowerSource entity.

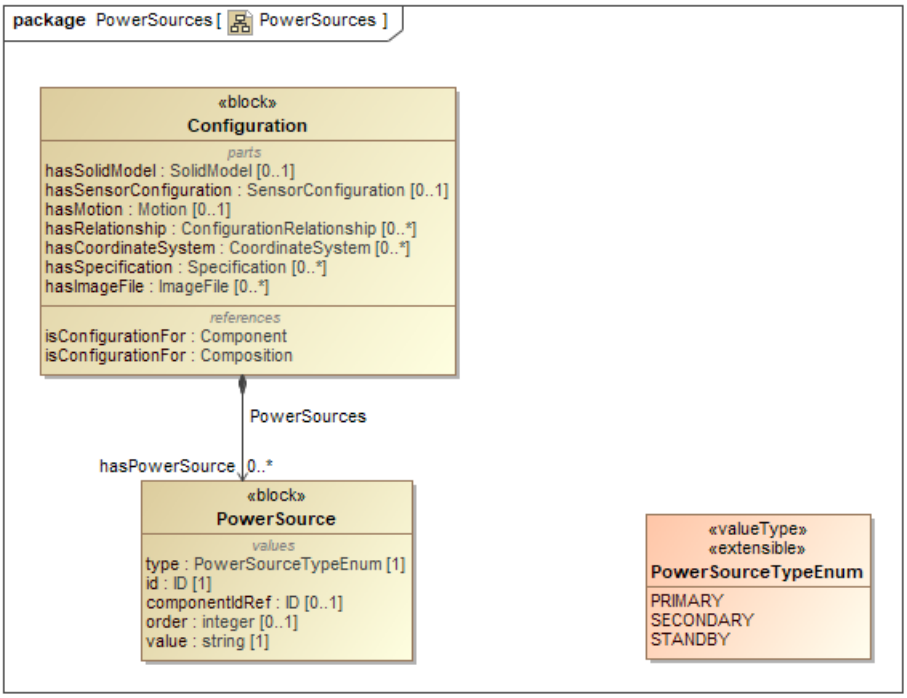

Figure 23: PowerSources

4169 8.9.1 PowerSource

4170 potential energy sources for the `Component`.

4171 The value of `PowerSource` **MUST** be `string`.

4172 8.9.1.1 Value Properties of PowerSource

4173 *Table 119* lists the Value Properties of `PowerSource`.

Value Property name	Value Property type	Multiplicity
`type`	`PowerSourceTypeEnum`	1
`id`	`ID`	1
`componentIdRef`	`ID`	0..1
`order`	`integer`	0..1

Table 119: Value Properties of PowerSource

4174 Descriptions for Value Properties of `PowerSource`:

- `type`
  type of the power source.
  `<<extensible>> PowerSourceTypeEnum` Enumeration:
  - `PRIMARY`
    main or principle.
  - `SECONDARY`
    alternate or not primary.
  - `STANDBY`
    held near at hand and ready for use and is uninterruptible.
- `id`
  unique identifier for the power source.
- `componentIdRef`
  reference to the `Component` providing observations about the power source.
- `order`
  optional precedence for a given power source.

9 Profile

MTConnect Profile is a *profile* that extends the Systems Modeling Language (SysML) metamodel for the MTConnect domain using additional data types and *stereotypes*.

9.1 DataTypes

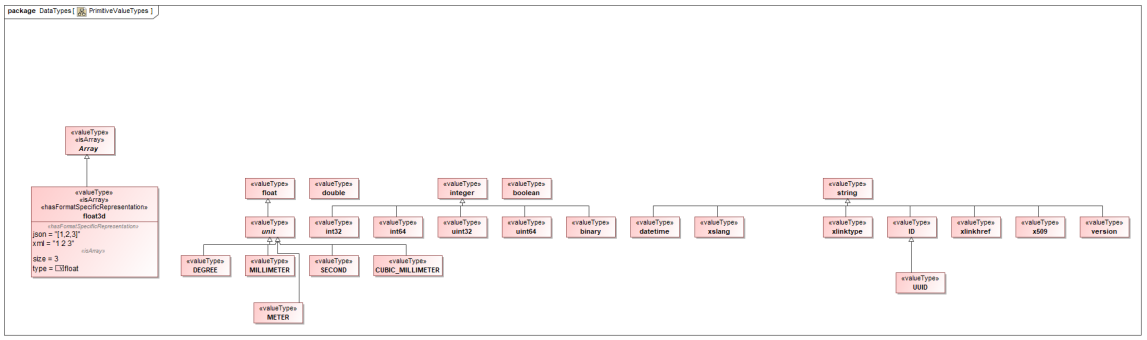

Figure 24: DataTypes

9.1.1 boolean

primitive type.

9.1.2 ID

string that represents an identifier (ID).

9.1.3 string

primitive type.

9.1.4 float

primitive type.

9.1.5 datetime

string that represents timestamp in ISO 8601 format.

9.1.6 integer

primitive type.

9.1.7 xlinktype

string that represents the type of an XLink element. See `https://www.w3.org/TR/xlink11/`.

9.1.8 xslang

string that represents a language tag. See `http://www.ietf.org/rfc/rfc4646.txt`.

9.1.9 SECOND

float that represents time in seconds.

9.1.10 xlinkhref

string that represents the locator attribute of an XLink element. See `https://www.w3.org/TR/xlink11/`.

9.1.11 x509

string that represents an `x509` data block. *Ref ISO/IEC 9594-8:2020*.

9.1.12 int32

32-bit integer.

9.1.13 int64

64-bit integer.

9.1.14 version

series of three numeric values, separated by a decimal point, representing a *major*, *minor*, and *patch* number of the MTConnect Standard.

9.1.15 uint32

32-bit unsigned integer.

9.1.16 uint64

64-bit unsigned integer.

9.1.17 binary

base-2 numeral system or binary numeral system represented by two digits: “0” and “1”.

9.1.18 double

primitive type.

9.1.19 Array

array.

9.1.20 <<hasFormatSpecificRepresentation>>float3d

array of size 3 and datatype float.

9.1.21 UUID

Universally Unique IDentifier. *Ref IETF:RFC-4122*

9.1.22 METER

float that represents measurement in meter.

9.2 Stereotypes

9.2.1 organizer

element that *organizes* other elements of a type.

9.2.2 deprecated

element that has been deprecated.

9.2.3 extensible

enumeration that can be extended.

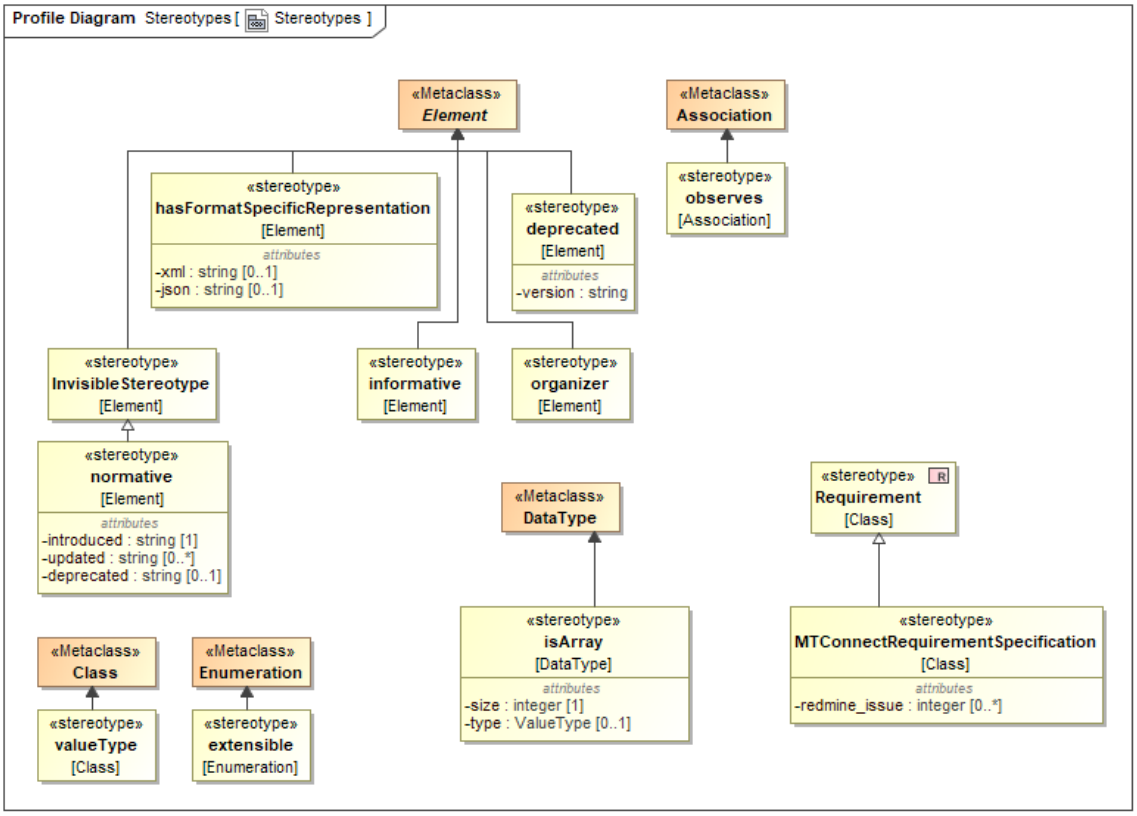

Figure 25: Stereotypes

9.2.4 informative

element that is descriptive and non-normative.

9.2.5 normative

element that has been added to the standard.

9.2.6 observes

association in which a *Component* makes *Observations* about an observable *DataItem*.

9.2.7 satisfiedBy

9.2.8 hasFormatSpecificRepresentation

element that has format specific representation that might be different from the element's SysML representation.

9.2.9 valueType

extends `Class`to be used as a SysML `<<ValueType>>`.

9.2.10 isArray

datatype that is an array.

9.2.11 MTConnectRequirementSpecification

MTConnect Requirement.

Appendices

A Bibliography

Engineering Industries Association. EIA Standard - EIA-274-D, Interchangeable Variable, Block Data Format for Positioning, Contouring, and Contouring/Positioning Numerically Controlled Machines. Washington, D.C. 1979.

ISO TC 184/SC4/WG3 N1089. ISO/DIS 10303-238: Industrial automation systems and integration Product data representation and exchange Part 238: Application Protocols: Application interpreted model for computerized numerical controllers. Geneva, Switzerland, 2004.

International Organization for Standardization. ISO 14649: Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers – Part 10: General process data. Geneva, Switzerland, 2004.

International Organization for Standardization. ISO 14649: Industrial automation systems and integration – Physical device control – Data model for computerized numerical controllers – Part 11: Process data for milling. Geneva, Switzerland, 2000.

International Organization for Standardization. ISO 6983/1 – Numerical Control of machines – Program format and definition of address words – Part 1: Data format for positioning, line and contouring control systems. Geneva, Switzerland, 1982.

Electronic Industries Association. ANSI/EIA-494-B-1992, 32 Bit Binary CL (BCL) and 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines. Washington, D.C. 1992.

National Aerospace Standard. Uniform Cutting Tests - NAS Series: Metal Cutting Equipment Specifications. Washington, D.C. 1969.

International Organization for Standardization. ISO 10303-11: 1994, Industrial automation systems and integration Product data representation and exchange Part 11: Description methods: The EXPRESS language reference manual. Geneva, Switzerland, 1994.

International Organization for Standardization. ISO 10303-21: 1996, Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure. Geneva, Switzerland, 1996.

H.L. Horton, F.D. Jones, and E. Oberg. Machinery's Handbook. Industrial Press, Inc.

New York, 1984.

International Organization for Standardization. ISO 841-2001: Industrial automation systems and integration - Numerical control of machines - Coordinate systems and motion nomenclature. Geneva, Switzerland, 2001.

ASME B5.57: Methods for Performance Evaluation of Computer Numerically Controlled Lathes and Turning Centers, 1998.

ASME/ANSI B5.54: Methods for Performance Evaluation of Computer Numerically Controlled Machining Centers. 2005.

OPC Foundation. OPC Unified Architecture Specification, Part 1: Concepts Version 1.00. July 28, 2006.

IEEE STD 1451.0-2007, Standard for a Smart Transducer Interface for Sensors and Actuators – Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH99684, October 5, 2007.

IEEE STD 1451.4-1994, Standard for a Smart Transducer Interface for Sensors and Actuators – Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, IEEE Instrumentation and Measurement Society, TC-9, The Institute of Electrical and Electronics Engineers, Inc., New York, N.Y. 10016, SH95225, December 15, 2004.

B XML Schema Diagrams

See XML schemas for the MTConnect standard here: https://schemas.mtconnect.org/.

B.1 Components Schema Diagrams

See Components element in MTConnectDevices schema.

See Description element in MTConnectDevices schema.

B.2 DataItems Schema Diagrams

See DataItems element in MTConnectDevices schema.

See Source element in MTConnectDevices schema.

See Constraints element in MTConnectDevices schema.

See Filter element in MTConnectDevices schema.

See Definition element in MTConnectDevices schema.

B.3 References Schema Diagrams

See References element in MTConnectDevices schema.

See ComponentRef element in MTConnectDevices schema.

See DataItemRef element in MTConnectDevices schema.

B.4 Configuration Schema Diagrams

See Configuration element in MTConnectDevices schema.

See CoordinateSystem element in MTConnectDevices schema.

See `Motion` element in `MTConnectDevices` schema.

See `Relationships` element in `MTConnectDevices` schema.

See `ComponentRelationship` element in `MTConnectDevices` schema.

See `DeviceRelationship` element in `MTConnectDevices` schema.

See `SensorConfiguration` element in `MTConnectDevices` schema.

See `SolidModel` element in `MTConnectDevices` schema.

See `Specifications` element in `MTConnectDevices` schema.

See `ProcessSpecification` element in `MTConnectDevices` schema.

C XML Examples

C.1 Device Entity Hierarchy Example

Example 3: Device Entity Hierarchy Example

```
<Devices>
  <Device id="d1" name="M12346" uuid="M80104K162N">
    <Components>
    <Axes id="a" name="base">
    <Components>
      <Linear id="x" name="X"/>
      <Linear id="y" name="Y"/>
      <Linear id="z" name="Z"/>
      <Rotary id="ar" name="B"/>
      <Linear id="w" name="Z3" nativeName="W"/>
      <Rotary id="c" name="C"/>
    </Components>
    </Axes>
    <Controller id="cont" name="controller">
    <Components>
      <Path id="path1" name="path"/>
    </Components>
    </Controller>
    <Door id="door1" name="door"/>
    <Systems id="systems" name="systems">
    <Components>
      <WorkEnvelope id="we1" name="workenv">
      <Compositions>
        <Composition type="WORKPIECE" id="wp"/>
```

```
25          <Composition type="TABLE" id="tbl"/>
26        </Compositions>
27        </WorkEnvelope>
28        <Electric id="elec" name="electric"/>
29        <Hydraulic id="hydraulic" name="hydraulic"/>
30        <Coolant id="coolant" name="coolant"/>
31        <Pneumatic id="pneumatic" name="pneumatic"/>
32        <Lubrication id="lubrication" name="lubrication"/>
33      </Components>
34      </Systems>
35      <Auxiliaries id="Aux" name="auxiliaries">
36      <Components>
37        <Environmental id="room" name="environmental"/>
38      </Components>
39      </Auxiliaries>
40      <Resources id="resources" name="resources">
41      <Components>
42        <Personnel id="personnel" name="personnel"/>
43        <Materials id="materials" name="materials">
44        <Components>
45          <Stock id="procstock" name="stock"/>
46        </Components>
47        <Materials/>
48      </Components>
49      </Resources>
50      </Components>
51    </Device>
52  </Devices>
```

C.2 Component Example

Example 4: Component Example

```
1   <Devices>
2     <Device id="d1" name="M12346" uuid="M80104K162N">
3       <Description manufacturer="Example\textunderscore␣Corporation"
4         serialNumber="272237"> Mill w/SMooth-G
5       </Description>
6     <DataItems>
7       <DataItem category="EVENT" id="avail" name="avail" type="
        AVAILABILITY"/>
8       <DataItem category="EVENT" id="d1\textunderscore␣asset\
            textunderscore␣chg" name="asset\textunderscore␣chg" type="
            ASSET\textunderscore␣CHANGED"/>
9       <DataItem category="EVENT" id="d1\textunderscore␣asset\
            textunderscore␣rem" name="asset\textunderscore␣rem" type="
            ASSET\textunderscore␣REMOVED"/>
10    </DataItems>
```

```
    <Components>
    <Controller id="cont" name="controller">
      <DataItems>
      <DataItem category="EVENT" id="estop" name="estop" type="
    EMERGENCY\textunderscore␣STOP"/>
      <DataItem category="CONDITION" id="logic\textunderscore␣cond"
          type="LOGIC\textunderscore␣PROGRAM"/>
      <DataItem category="CONDITION" id="cont\textunderscore␣system"
           type="SYSTEM"/>
      <DataItem category="SAMPLE" id="cut\textunderscore␣time" type=
          "ACCUMULATED\textunderscore␣TIME" units="SECOND"/>
    </DataItems>
    <Components>
      <Path id="path1" name="path">
        <DataItems>
        <DataItem category="EVENT" id="execution" name="execution"
    type="EXECUTION"/>
        <DataItem category="SAMPLE" id="cspeed" subType="ACTUAL"
           type="CUTTING\textunderscore␣SPEED" units="MILLIMETER/
           SECOND"/>
        <DataItem category="CONDITION" id="path\textunderscore␣
           system" type="SYSTEM"/>
        <DataItem category="EVENT" id="cvars" representation="DATA\
           textunderscore␣SET" type="VARIABLE"/>
        <DataItem category="EVENT" id="woffset" representation="
           TABLE" type="WORK\textunderscore␣OFFSET"/>
      </DataItems>
      </Path>
    </Components>
    </Controller>
  </Components>
  </Device>
</Devices>
```

C.3 Component with Compositions Example

In XML, `Composition` types are represented differently than `Component` types. For `Component` types, the element name is Pascal Case of the `Component` type name. Whereas, the element name for all `Composition`types is `Composition` and the type is defined by the `type` attribute of the element (see example below).

Example 5: Component with Compositions Example

```
<WorkEnvelope id="we1" name="workenv">
  <DataItems>
    <DataItem category="CONDITION" id="hardware1" name="hardware1"
    type="HARDWARE"/>
  </DataItems>
```

```
    <Compositions>
    <Composition type="WORKPIECE" id="wp"/>
    <Composition type="TABLE" id="tbl"/>
    </Compositions>
</WorkEnvelope>
```

C.4 CoordinateSystem Example

Example 6: CoordinateSystem Example

```
<Devices>
  <Device id="d1" name="M12346" uuid="M80104K162N">
    <Description manufacturer="Example\textunderscore␣Corporation"
      serialNumber="272237"> Mill w/SMooth-G
    </Description>
    <DataItems>
      <DataItem id="avail" type="AVAILABILITY" category="EVENT"/>
      <DataItem category="EVENT" id="d1\textunderscore␣asset\
          textunderscore␣chg" name="asset\textunderscore␣chg" type="
          ASSET\textunderscore␣CHANGED"/>
      <DataItem category="EVENT" id="d1\textunderscore␣asset\
          textunderscore␣rem" name="asset\textunderscore␣rem" type="
          ASSET\textunderscore␣REMOVED"/>
    </DataItems>
    <Configuration>
      <CoordinateSystems>
        <CoordinateSystem id="base" type="BASE">
          <Origin>0 0 0</Origin>
        </CoordinateSystem>
        <CoordinateSystem id="machcoord" type="MACHINE" parentIdRef=
            "base">
          <Transformation>
            <Translation>210 275 1430</Translation>
            <Rotation>0 0 0</Rotation>
          </Transformation>
        </CoordinateSystem>
        <CoordinateSystem id="workcoord" type="OBJECT" parentIdRef="
            machcoord">
          <Transformation>
            <Translation>0 0 0</Translation>
            <Rotation>0 0 0</Rotation>
          </Transformation>
        </CoordinateSystem>
      </CoordinateSystems>
    </Configuration>
  </Device>
</Devices>
```

C.5 Motion Example

Example 7: Motion Example

```
<Devices>
  <Device id="d1" name="M12346" uuid="M80104K162N">
    <Description manufacturer="Example\textunderscore␣Corporation"
      serialNumber="272237"> Mill w/SMooth-G
    </Description>
    <DataItems>
      <DataItem id="avail" type="AVAILABILITY" category="EVENT"/>
      <DataItem category="EVENT" id="d1\textunderscore␣asset\
         textunderscore␣chg" name="asset\textunderscore␣chg" type="
         ASSET\textunderscore␣CHANGED"/>
      <DataItem category="EVENT" id="d1\textunderscore␣asset\
         textunderscore␣rem" name="asset\textunderscore␣rem" type="
         ASSET\textunderscore␣REMOVED"/>
    </DataItems>
    <Configuration>
      <CoordinateSystems>
        <CoordinateSystem id="base" type="BASE">
          <Origin>0 0 0</Origin>
        </CoordinateSystem>
        <CoordinateSystem id="machcoord" type="MACHINE" parentIdRef=
           "base">
          <Transformation>
            <Translation>210 275 1430</Translation>
            <Rotation>0 0 0</Rotation>
          </Transformation>
        </CoordinateSystem>
        <CoordinateSystem id="workcoord" type="OBJECT" parentIdRef="
           machcoord">
          <Transformation>
            <Translation>0 0 0</Translation>
            <Rotation>0 0 0</Rotation>
          </Transformation>
        </CoordinateSystem>
      </CoordinateSystems>
    </Configuration>
    <Components>
      <Axes id="a" name="base">
        <Components>
          <Linear id="x" name="X">
            <Configuration>
              <Motion id="xmotion" coordinateSystemIdRef="machcoord"
     type="PRISMATIC" actuation="DIRECT">
                <Transformation>
                  <Translaton>300 915 590</Translaton>
                  <Rotation>0 0 0</Rotation>
                </Transformation>
                <Axis>1.0 0 0</Axis>
```

```
              </Motion>
            </Configuration>
          </Linear>
          <DataItems>
            <DataItem id="xtravel" type="SYSTEM" category="CONDITION
    ">
          </DataItems>
        </Components>
      </Axes>
    </Components>
  </Device>
</Devices>
```

C.6 Relationship Example

Example 8: Relationship Example

```
<Components>
  <Axes id="a" name="base">
    <Components>
      <Linear id="x" name="X">
        <Configuration>
          <Relationships>
            <ComponentRelationship id="xpar" type="PARENT" idRef="
    we1"/>
          </Relationships>
        </Configuration>
      </Linear>
      <DataItems>
        <DataItem id="xtravel" type="SYSTEM" category="CONDITION">
      </DataItems>
    </Components>
  </Axes>
  <Systems id="systems" name="systems">
    <Components>
      <WorkEnvelope id="we1" name="workenv">
        <Compositions>
          <Composition type="WORKPIECE" id="wp"/>
          <Composition type="TABLE" id="tbl"/>
        </Compositions>
        <DataItems>
          <DataItem id="we1cond" type="SYSTEM" category="CONDITION">
        </DataItems>
      </WorkEnvelope>
    </Components>
  </Systems>
</Components>
```

C.7 SolidModel Example

Example 9: SolidModel Example

```
<Devices>
  <Device id="d1" name="M12346" uuid="M80104K162N">
    <Description manufacturer="Example\textunderscore␣Corporation"
      serialNumber="272237"> Mill w/SMooth-G
    </Description>
    <DataItems>
      <DataItem id="avail" type="AVAILABILITY" category="EVENT"/>
      <DataItem category="EVENT" id="d1\textunderscore␣asset\
         textunderscore␣chg" name="asset\textunderscore␣chg" type="
         ASSET\textunderscore␣CHANGED"/>
      <DataItem category="EVENT" id="d1\textunderscore␣asset\
         textunderscore␣rem" name="asset\textunderscore␣rem" type="
         ASSET\textunderscore␣REMOVED"/>
    </DataItems>
    <Configuration>
      <CoordinateSystems>
        <CoordinateSystem id="base" type="BASE">
          <Origin>0 0 0</Origin>
        </CoordinateSystem>
        <CoordinateSystem id="machcoord" type="MACHINE" parentIdRef=
           "base">
          <Transformation>
            <Translation>210 275 1430</Translation>
            <Rotation>0 0 0</Rotation>
          </Transformation>
        </CoordinateSystem>
        <CoordinateSystem id="workcoord" type="OBJECT" parentIdRef="
           machcoord">
          <Transformation>
            <Translation>0 0 0</Translation>
            <Rotation>0 0 0</Rotation>
          </Transformation>
        </CoordinateSystem>
      </CoordinateSystems>
    </Configuration>
    <Components>
      <Structures id="struct">
        <Configuration>
          <SolidModel id="model" mediaType="OBJ" href="/objs/mazak.
    obj" coordinateSystemIdRef="base">
            <Transformation>
              <Translation>0 860 0</Translation>
              <Rotation>-90 0 0</Rotation>
            </Transformation>
            <Scale>0.001 0.001 0.001</Scale>
          </SolidModel>
        </Configuration>
```

```
        <Components>
          <Structure id="xaxis" name="X\textunderscore␣AXIS">
            <Configuration>
              <SolidModel id="x\textunderscore␣model" mediaType="OBJ
    " itemRef="xaxis" solidModelIdRef="model" coordinateSystemIdRef="
    base"/>
            </Configuration>
            <DataItems>
              <DataItem type="SYSTEM" category="CONDITION" id="
    struct\textunderscore␣cond"/>
            </DataItems>
          </Structure>
        </Components>
      </Structures>
    </Components>
  </Device>
</Devices>
```

C.8 Specification Example

Example 10: Specification Example

```
<Components>
  <Axes id="a" name="base">
    <Components>
      <Linear id="x" name="X">
        <Configuration>
          <Specifications>
            <Specification id="spec001" type="LENGTH" units="
    MILLIMETER">
              <Nominal>650</Nominal>
            </Specification>
            <Specification id="spec002" type="LINEAR\textunderscore␣
                FORCE" units="NEWTON">
              <Maximum>5200</Maximum>
              <Minimum>0</Minimum>
            </Specification>
          </Specifications>
        </Configuration>
        <DataItems>
          <DataItem id="xtravel" type="SYSTEM" category="CONDITION">
        </DataItems>
      </Linear>
    </Components>
  </Axes>
</Components>
```

C.9 Example of sensing element provided as data item associated with a Component

Example 11: Example of sensing element provided as data item associated with a Component

```
<Components>
  <Axes id="a" name="base"
    <Components>
      <Rotary id="c" name="C">
        <DataItems>
          <DataItem type="TEMPERATURE" id="servotemp4"
            category="SAMPLE" nativeUnits="CELSIUS"
            compositionId="Bmotor" units="CELSIUS"/>
          </DataItems>
      </Rotary>
    </Components>
  </Axes>
</Components>
```

C.10 Example of Sensor for rotary axis

Example 12: Example of Sensor for rotary axis

```
<Components>
  <Axes id="a" name="base"
    <Components>
      <Rotary id="ar" name="B">
        <Components>
          <Sensor id="spdlm" name="Spindlemonitor">
            <DataItems>
              <DataItem type="DISPLACEMENT" id="cvib"
                category="SAMPLE" name="Svib"
                units="MILLIMETER"/>
            </DataItems>
          </Sensor >
        <Components>
      </Rotary>
    </Components>
  </Axes>
</Components>
```

C.11 Example of sensor unit with sensing element

Example 13: Example of sensor unit with sensing element

```
<Axes id="a" name="base"
  <Components>
  <Sensor id="sens1" name="Sensorunit">
    <DataItems>
    <DataItem type="TEMPERATURE" id="sentemp"
      category="SAMPLE" name="Sensortemp"
      units="CELSIUS"/>
    </DataItems>
  </Sensor >
  <Linear id="x" name="X">
    <DataItems>
    <DataItem type="DISPLACEMENT" id="xvib"
      category="SAMPLE" name="xvib"
      units="MILLIMETER">
      <Source componentId="sens1"/>
    </DataItem>
    </DataItems>
  </Rotary>
  <Linear id="y" name="Y">
    <DataItems>
    <DataItem type="DISPLACEMENT" id="yvib"
      category="SAMPLE" name="yvib"
      units="MILLIMETER">
      <Source componentId="sens1"/>
    </DataItem>
    </DataItems>
  </Linear>
  <Components>
</Axes>
```

C.12 Example of configuration data for Sensor

Example 14: Example of configuration data for Sensor

```
<Sensor id="sensor" name="sensor">
  <Configuration>
    <SensorConfiguration>
      <FirmwareVersion>2.02</FirmwareVersion>
      <CalibrationDate>2010-05-16</CalibrationDate>
      <NextCalibrationDate>2010-05-16</NextCalibrationDate>
      <CalibrationInitials>WS</CalibrationInitials>
      <Channels>
        <Channel number="1" name="A/D:1">
          <Description>A/D With Thermister</Description>
        </Channel>
      </Channels>
    </SensorConfiguration>
```

```
4757 14    </Configuration>
4758 15    <DataItems>
4759 16      <DataItem category="CONDITION" id="sensorc"
4760 17        name="sensorc" type="SYSTEM" />
4761 18      <DataItem category="SAMPLE" id="senv" name="sensorc"
4762 19        type="VOLTAGE\textunderscore␣DC" units="VOLT" subType="ACTUAL"
4763             />
4764 20    </DataItems>
4765 21  </Sensor>
```