MTConnect® Standard
Part 2.0 – Device Information Model
Version 2.2.0

MTConnect Specification and Materials

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*.

14 2 Terminology and Conventions

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

17 2.1 MTConnect References

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

19 [MTConnect Part 2.0] *MTConnect Standard: Part 2.0 - Device Information Model*. Ver-
20 sion 2.0.

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

23 [MTConnect Part 4.0] *MTConnect Standard: Part 4.0 - Asset Information Model*. Ver-
24 sion 2.0.

25

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 sub-parts 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.

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

3.1 Device

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

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

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

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

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

72 See *Section 3.1.2 - Part Properties of Device* for a list of *top level* `Compo-`
73 `nent` types for a `Device`.

3.1.1 Value Properties of Device

75 *Table 1* lists the Value Properties of `Device`.

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

Table 1: Value Properties of Device

76 Descriptions for Value Properties of `Device`:

77 • <<deprecated>> iso841Class

78   **DEPRECATED** in *MTConnect Version 1.2*.

79   • `mtconnectVersion`

80   MTConnect version of the *Device Information Model* used to configure the informa-
81   tion to be published for a piece of equipment in an *MTConnect Response Document*.

82   • `hash`

83   condensed message digest from a secure one-way hash function. *Ref FIPS PUB*
84   *180-4*

85 3.1.2   Part Properties of Device

86 *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

87 Descriptions for Part Properties of `Device`:

88   • `Auxiliary`

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

91   • `Controller`

92   `System` that provides regulation or management of a system or component. *Ref ISO*
93   *16484-5:2017*

94   • `Interface`

95   abstract `Component` that coordinates actions and activities between pieces of equip-
96   ment.

97 • `Resource`

98   abstract `Component` composed of material or personnel involved in a manufactur-
99   ing process.

100   • `Structure`

101   `Component` composed of part(s) comprising the rigid bodies of the piece of equip-
102   ment.

103   • `System`

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

105   • `Axis`

106   abstract `Component` composed of a motion system that provides linear or rota-
107   tional motion for a piece of equipment.

108   • `Adapter`

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

111 3.1.3   Commonly Observed DataItem Types for Device

112 *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 equip-
ment. Figure 1 provides an overview of the entities used in an example of an MTCon-
nectDevices entity.
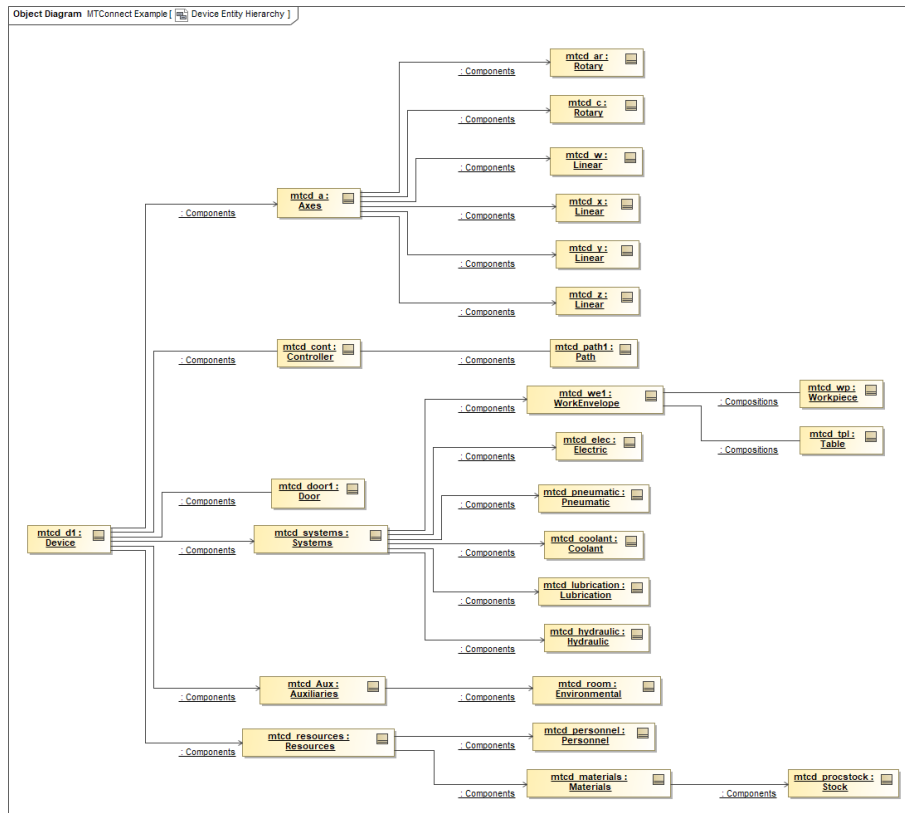
Figure 1: Device Entity Hierarchy Example

Note 1 to entry: See *Example 1* for the Extensible Markup Language (XML)
representation of the same example.

Note 2 to entry: Example above only shows the Component and Composi-
tion 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.

125 The first, or highest level, entity defined for an `MTConnectDevices` entity is `De-`
126 `vices`. `Devices` is used to group one or more pieces of equipment into a single docu-
127 ment. `Devices` **MUST** always be defined for an `MTConnectDevices` entity.

128 `Device` is the next entity defined for the `MTConnectDevices` entity. A separate `De-`
129 `vice` entity is used to identify each piece of equipment for an `MTConnectDevices`
130 entity. Each `Device` provides information on the physical and logical structure of the
131 piece of equipment and the data associated with that equipment. `Device` can also repre-
132 sent any logical grouping of pieces of equipment that function as a unit or any other data
133 source that provides data through an *agent*.

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

136 `Components` is the next entity defined for the `MTConnectDevices` entity. `Com-`
137 `ponents` is used to group information describing lower level physical parts or logical
138 functions of a piece of equipment.

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

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

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

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

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

155 Note: See Figure 1 for an example.

156 4.1   Components

157 This section provides semantic information for the `Component` entity. Figure 2 shows
158 the `Component` model.
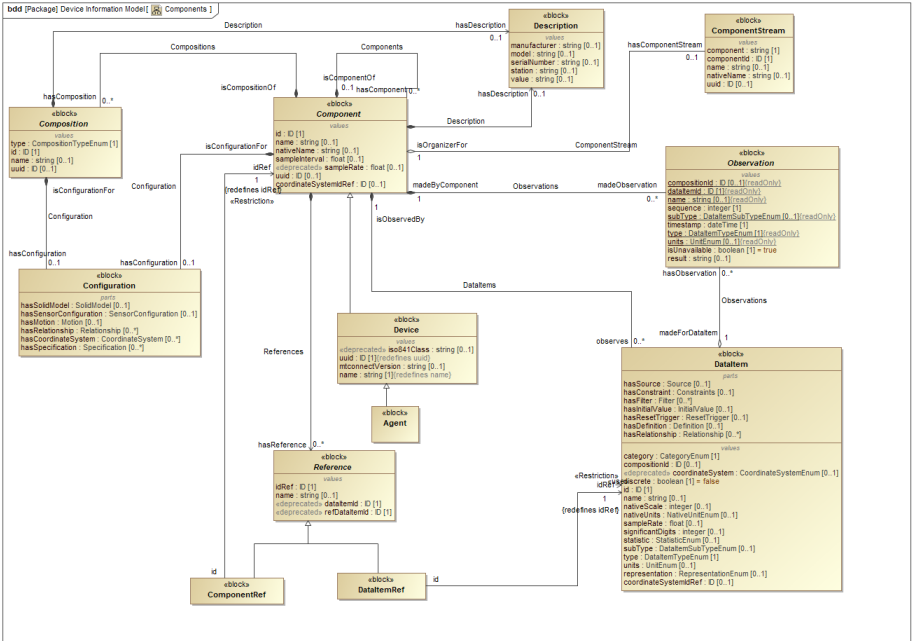
Figure 2: Components

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

160 4.1.1   Component

161 logical or physical entity that provides a capability.

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

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

167 At least one of `Component`, `DataItem`, or `Reference` entities **MUST** be provided
168 for a `Component`.
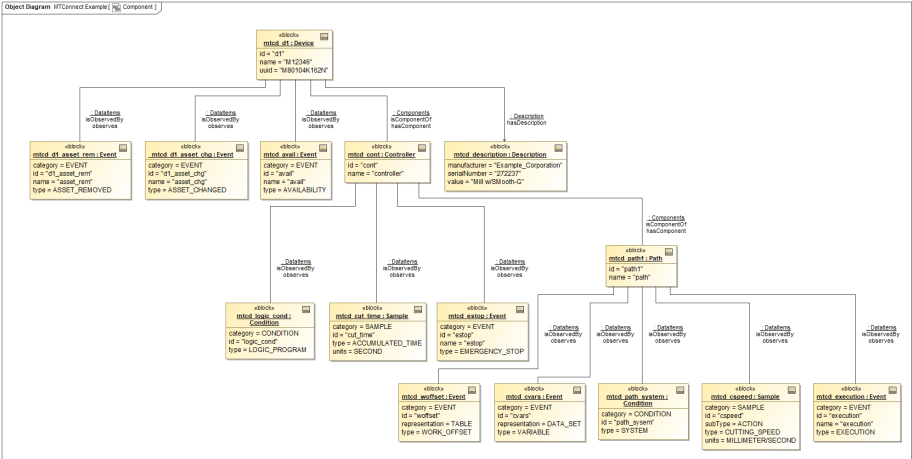
Figure 3: Component Example

169   Note: See *Example 2* for the XML representation of the same example.

170 **4.1.1.1   Value Properties of Component**

171 *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`	`ID`	0..1
`coordinateSystemIdRef`	`ID`	0..1

Table 4: Value Properties of Component

172 Descriptions for Value Properties of `Component`:

173   • `id`

174   unique identifier for the `Component`.

175   • `name`

176   name of the `Component`.

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

178   parent `Component`.

179   • `nativeName`

180   common name associated with `Component`.

181   • `sampleInterval`

182   interval in milliseconds between the completion of the reading of the data associated

183   with the `Component` until the beginning of the next sampling of that data.

184   This information may be used by client software applications to understand how

185   often information from a `Component` is expected to be refreshed.

186   The refresh rate for data from all child `Component` entities will be the same as for

187   the parent `Component` element unless specifically overridden by another `sam-`

188   `pleInterval` provided for the child `Component`.

189   • `<<deprecated>> sampleRate`

190   **DEPRECATED** in *MTConnect Version 1.2*. Replaced by `sampleInterval`, `Component`.

191   • `uuid`

192   universally unique identifier for the `Component`.

193   • `coordinateSystemIdRef`

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

195 **4.1.1.2   Reference Properties of Component**

196 *Table 5* lists the Reference Properties of `Component`.

Reference Property name	Multiplicity
`ComponentStream`	0..1

Table 5: Reference Properties of Component

197 Descriptions for Reference Properties of `Component`:

198   • `ComponentStream`

199   organizes the data associated with each `Component` entity defined for a `Device`

200   in the associated *MTConnectDevices Response Document*.

201   `ComponentStream` *organizes* the `Observation` entities associated with the

202   `Component`.

203   See *MTConnect Standard: Part 3.0 - Observation Information Model* for the `Com-`

204   `ponentStream` model.

205 Note 1 to entry: In the XML representation, `ComponentStream` enti-
206 ties **MUST NOT** appear in the *MTConnectDevices Response Document*.

207 Note 2 to entry: In the XML representation, `ComponentStream` enti-
208 ties **MUST** appear only in the *MTConnectStreams Response Document*.

4.1.1.3 Part Properties of Component

210 *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

211 Descriptions for Part Properties of `Component`:

212   • `Description`
213   descriptive content.
214   See *Section 4.1.2 - Description*.

215   • `Composition`
216   functional part of a piece of equipment contained within a `Component`.
217   `Compositions` groups one or more `Composition` entities. See *Section 5.1 -*
218   *Compositions*.

219   • `Component`
220   logical or physical entity that provides a capability.
221   `Components` groups one or more `Component` entities.

222   • `Configuration`
223   technical information about an entity describing its physical layout, functional char-
224   acteristics, and relationships with other entities.
225   See *Section 8.1 - Configurations*.

226 • `DataItem`

227 information reported about a piece of equipment.

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

229 • `Observation`

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

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

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

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

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

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

240 • `Reference`

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

243 `References` groups one or more `Reference` entities associated with the `Com-`
244 `ponent`. See *Section 7.1 - References*.

245 **4.1.2 Description**

246 descriptive content.

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

248 Note 2 to entry: See *Example 2* for the XML representation of the same ex-
249 ample.

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

251 **4.1.2.1 Value Properties of Description**

252 *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

253 Descriptions for Value Properties of `Description`:

254 • manufacturer

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

257 • model

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

260 • serialNumber

261 serial number associated with a piece of equipment.

262 • station

263 station where the physical part or logical function of a piece of equipment is located
264 when it is part of a manufacturing unit or cell with multiple stations.

265 4.2 Devices

266 This section provides semantic information for the `Device` types.

267 4.2.1 Agent

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

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

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

271 An `Agent` **MUST** provide connection information for each data source currently supply-
272 ing data to the *MTConnect Agent*.

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

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

275 4.3   Component Types

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

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

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

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

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

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

292 4.3.1   Actuator

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

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

297 4.3.2 Adapter

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

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

301 4.3.3 Amplifier

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

304 4.3.3.1 Part Properties of Amplifier

305 *Table 8* lists the Part Properties of `Amplifier`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 8: Part Properties of Amplifier

306 Descriptions for Part Properties of `Amplifier`:

307 • `Composition`

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

309 • `Component`

310 logical or physical entity that provides a capability.

311 4.3.4 Auxiliary

312 abstract `Component` composed of removable part(s) of a piece of equipment that pro-
313 vides supplementary or extended functionality.

314 4.3.4.1 Deposition

315 `Auxiliary` that manages the addition of material or state change of material being per-
316 formed in an additive manufacturing process.

317 For example, this could describe the portion of a piece of equipment that manages a mate-
318 rial extrusion process or a vat polymerization process.

319 4.3.4.2 Loader

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

322 4.3.4.2.1 BarFeeder

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

324 4.3.4.3 ToolingDelivery

325 `Auxiliary` that manages, positions, stores, and delivers tooling within a piece of equip-
326 ment.

327 4.3.4.3.1 GangToolBar

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

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

332 4.3.4.3.2 AutomaticToolChanger

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

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

337 **4.3.4.3.3 ToolMagazine**

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

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

342 **4.3.4.3.4 ToolRack**

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

345 Tools are located in `Station` entities.

346 **4.3.4.3.5 Turret**

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

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

350 **4.3.4.4 WasteDisposal**

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

352 4.3.5 Axis

353 abstract `Component` composed of a motion system that provides linear or rotational mo-
354 tion for a piece of equipment.

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

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

360 MTConnect has two high-level classes for automation equipment as follows: (1) Equip-

361 ment that controls cartesian coordinate axes and (2) Equipment that controls articulated
362 axes. There are ambiguous cases where some machines exhibit both characteristics; when
363 this occurs, the primary control system's configuration determines the classification.

364 Examples of cartesian coordinate equipment are CNC Machine Tools, Coordinate mea-
365 surement machines, as specified in ISO 841, and 3D Printers. Examples of articulated
366 automation equipment are Robotic systems as specified in ISO 8373.

367 The following sections define the designation of names for the axes and additional guid-
368 ance when selecting the correct scheme to use for a given piece of equipment.

4.3.5.1   Cartesian Coordinate Naming Conventions

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

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

4.3.5.1.1   Linear Motion

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

4.3.5.1.2   Rotary Motion

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

4.3.5.2   Articulated Machine Control Systems

385 An articulated control system's axes represent the connecting linkages between two ad-
386 jacent rigid members of an assembly. The `Linear` axis represents prismatic motion,
387 and the `Rotary` axis represents the rotational motion of the two related members. The

388 control organizes the axes in a kinematic chain from the mounting surface (base) to the
389 end-effector or tooling.

4.3.5.3 Articulated Machine Axis Names

391 The axes of articulated machines represent forward kinematic relationships between me-
392 chanical linkages. Each axis is a connection between linkages, also referred to as joints,
393 and **MUST** be named using a `J` followed by a monotonically increasing number; for ex-
394 ample, `J1`, `J2`, `J3`. The numbering starts at the base axis connected or closest to the
395 mounting surface, `J1`, incrementing to the mechanical interface, `Jn`, where `n` is the num-
396 ber of the last axis. The chain forms a parent-child relationship with the parent being the
397 axis closest to the base.

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

4.3.5.4 Linear

404 `Axis` that provides prismatic motion along a fixed axis.

4.3.5.5 Part Properties of Linear

406 *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

407 Descriptions for Part Properties of `Linear`:

408   • `Load`

409   actual versus the standard rating of a piece of equipment.

410　　• Temperature

411　　　　degree of hotness or coldness measured on a definite scale.

412　　• AxisFeedrate.Actual

413　　　　measured or reported value of an observation.

414 **4.3.5.6　Rotary**

415 `Axis` that provides rotation about a fixed axis.

416 **4.3.5.7　Part Properties of Rotary**

417 *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

418 Descriptions for Part Properties of `Rotary`:

419　　• Load

420　　　　actual versus the standard rating of a piece of equipment.

421　　• Temperature

422　　　　degree of hotness or coldness measured on a definite scale.

423　　• RotaryVelocity

424　　　　rotational speed of a rotary axis.

425　　• AxisFeedrate

426　　　　feedrate of a linear axis.

427 4.3.5.8 `<<deprecated>>`Spindle

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

430 `Spindle` was **DEPRECATED** in *MTConnect Version 1.1* and was replaced by `Ro-`
431 `taryMode.`

432 4.3.6 Ballscrew

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

435 4.3.6.1 Part Properties of Ballscrew

436 *Table 11* lists the Part Properties of `Ballscrew`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 11: Part Properties of Ballscrew

437 Descriptions for Part Properties of `Ballscrew`:

438 • `Composition`

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

440 • `Component`

441 logical or physical entity that provides a capability.

442 4.3.7 Belt

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

4.3.7.1 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.

4.3.8 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.

4.3.8.1 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.

462 4.3.9 Chain

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

465 4.3.9.1 Part Properties of Chain

466 *Table 14* lists the Part Properties of `Chain`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 14: Part Properties of Chain

467 Descriptions for Part Properties of `Chain`:

468 • `Composition`
469   functional part of a piece of equipment contained within a `Component`.

470 • `Component`
471   logical or physical entity that provides a capability.

472 4.3.10 Chopper

473 leaf `Component` that breaks material into smaller pieces.

474 4.3.10.1 Part Properties of Chopper

475 *Table 15* lists the Part Properties of `Chopper`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 15: Part Properties of Chopper

476 Descriptions for Part Properties of `Chopper`:

477 • `Composition`

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

479 • `Component`

480   logical or physical entity that provides a capability.

481   4.3.11   Chuck

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

483   4.3.12   Chute

484 leaf `Component` composed of an inclined channel that conveys material.

485   4.3.12.1   Part Properties of Chute

486 *Table 16* lists the Part Properties of `Chute`.

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

Table 16: Part Properties of Chute

487 Descriptions for Part Properties of `Chute`:

488 • `Composition`

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

490 • `Component`

491   logical or physical entity that provides a capability.

492   4.3.13   CircuitBreaker

493 leaf `Component` that interrupts an electric circuit.

494 **4.3.13.1 Part Properties of CircuitBreaker**

495 *Table 17* lists the Part Properties of `CircuitBreaker`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 17: Part Properties of CircuitBreaker

496 Descriptions for Part Properties of `CircuitBreaker`:

497 • `Composition`

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

499 • `Component`

500 logical or physical entity that provides a capability.

501 **4.3.14 Clamp**

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

503 **4.3.14.1 Part Properties of Clamp**

504 *Table 18* lists the Part Properties of `Clamp`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 18: Part Properties of Clamp

505 Descriptions for Part Properties of `Clamp`:

506 • `Composition`

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

508 • `Component`

509 logical or physical entity that provides a capability.

510 4.3.15 Compressor

511 leaf `Component` composed of a pump or other mechanism that reduces volume and in-
512 creases pressure of gases in order to condense the gases to drive pneumatically powered
513 pieces of equipment.

514 4.3.15.1 Part Properties of Compressor

515 *Table 19* lists the Part Properties of `Compressor`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 19: Part Properties of Compressor

516 Descriptions for Part Properties of `Compressor`:

517 • `Composition`
518 functional part of a piece of equipment contained within a `Component`.

519 • `Component`
520 logical or physical entity that provides a capability.

521 4.3.16 CoolingTower

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

524 4.3.16.1 Part Properties of CoolingTower

525 *Table 20* lists the Part Properties of `CoolingTower`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 20: Part Properties of CoolingTower

526 Descriptions for Part Properties of `CoolingTower`:

527 • `Composition`

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

529 • `Component`

530 logical or physical entity that provides a capability.

531 4.3.17 Door

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

535 The closure can be opened or closed to allow or restrict access to other parts of the equip-
536 ment.

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

539 4.3.17.1 Commonly Observed DataItem Types for Door

540 *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

541 4.3.18   Drain

542 leaf `Component` that allows material to flow for the purpose of drainage from, for exam-
543 ple, a vessel or tank.

544 4.3.18.1   Part Properties of Drain

545 *Table 22* lists the Part Properties of `Drain`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 22: Part Properties of Drain

546 Descriptions for Part Properties of `Drain`:

547 • `Composition`
548   functional part of a piece of equipment contained within a `Component`.

549 • `Component`
550   logical or physical entity that provides a capability.

551 4.3.19   Encoder

552 leaf `Component` that measures position.

553 4.3.19.1   Part Properties of Encoder

554 *Table 23* lists the Part Properties of `Encoder`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 23: Part Properties of Encoder

555 Descriptions for Part Properties of `Encoder`:

556   • Composition

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

558   • Component

559   logical or physical entity that provides a capability.

4.3.20   Environmental

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

562   Note: `Environmental` **SHOULD** be organized by `Auxillaries`, `Sys-`
563   `tems` or `Parts` depending on the relationship to the `Component`.

4.3.21   ExpiredPot

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

4.3.21.1   Part Properties of ExpiredPot

568 *Table 24* lists the Part Properties of `ExpiredPot`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 24: Part Properties of ExpiredPot

569 Descriptions for Part Properties of `ExpiredPot`:

570   • Composition

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

572   • Component

573   logical or physical entity that provides a capability.

574 ## 4.3.22 ExposureUnit

575 leaf `Component` that emits a type of radiation.

576 ### 4.3.22.1 Part Properties of ExposureUnit

577 *Table 25* lists the Part Properties of `ExposureUnit`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 25: Part Properties of ExposureUnit

578 Descriptions for Part Properties of `ExposureUnit`:

579 • `Composition`
580 functional part of a piece of equipment contained within a `Component`.

581 • `Component`
582 logical or physical entity that provides a capability.

583 ## 4.3.23 ExtrusionUnit

584 leaf `Component` that dispenses liquid or powered materials.

585 ### 4.3.23.1 Part Properties of ExtrusionUnit

586 *Table 26* lists the Part Properties of `ExtrusionUnit`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 26: Part Properties of ExtrusionUnit

587 Descriptions for Part Properties of `ExtrusionUnit`:

588   • Composition
589   functional part of a piece of equipment contained within a `Component`.

590   • Component
591   logical or physical entity that provides a capability.

592 4.3.24 Fan

593 leaf `Component` that produces a current of air.

594 4.3.24.1 Part Properties of Fan

595 *Table 27* lists the Part Properties of `Fan`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 27: Part Properties of Fan

596 Descriptions for Part Properties of `Fan`:

597   • Composition
598   functional part of a piece of equipment contained within a `Component`.

599   • Component
600   logical or physical entity that provides a capability.

601 4.3.25 Filter

602 leaf `Component` through which liquids or gases are passed to remove suspended impuri-
603 ties or to recover solids.

604 4.3.25.1 Part Properties of Filter

605 *Table 28* lists the Part Properties of `Filter`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 28: Part Properties of Filter

606 Descriptions for Part Properties of `Filter`:

607 • `Composition`
608   functional part of a piece of equipment contained within a `Component`.

609 • `Component`
610   logical or physical entity that provides a capability.

611 **4.3.26 Galvanomotor**

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

614 **4.3.26.1 Part Properties of Galvanomotor**

615 *Table 29* lists the Part Properties of `Galvanomotor`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 29: Part Properties of Galvanomotor

616 Descriptions for Part Properties of `Galvanomotor`:

617 • `Composition`
618   functional part of a piece of equipment contained within a `Component`.

619 • `Component`
620   logical or physical entity that provides a capability.

621 **4.3.27 Gripper**

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

623 **4.3.27.1 Part Properties of Gripper**

624 *Table 30* lists the Part Properties of `Gripper`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 30: Part Properties of Gripper

625 Descriptions for Part Properties of `Gripper`:

626 • `Composition`
627   functional part of a piece of equipment contained within a `Component`.
628 • `Component`
629   logical or physical entity that provides a capability.

630 **4.3.28 Hopper**

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

633 **4.3.28.1 Part Properties of Hopper**

634 *Table 31* lists the Part Properties of `Hopper`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 31: Part Properties of Hopper

635 Descriptions for Part Properties of `Hopper`:

636 • `Composition`

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

638 • `Component`

639 logical or physical entity that provides a capability.

4.3.29 LinearPositionFeedback

641 leaf `Component` that measures linear motion or position.

642 **DEPRECATION WARNING** : May be deprecated in the future. Recommend using
643 `Encoder`.

4.3.29.1 Part Properties of LinearPositionFeedback

645 *Table 32* lists the Part Properties of `LinearPositionFeedback`.

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

Table 32: Part Properties of LinearPositionFeedback

646 Descriptions for Part Properties of `LinearPositionFeedback`:

647 • `Composition`

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

649 • `Component`

650 logical or physical entity that provides a capability.

4.3.30 Lock

652 `Component` that physically prohibits a `Device` or `Component` from opening or oper-
653 ating.

654 **4.3.30.1   Commonly Observed DataItem Types for Lock**

655 *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

656   4.3.31   Motor

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

659   4.3.31.1   Part Properties of Motor

660 *Table 34* lists the Part Properties of `Motor`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 34: Part Properties of Motor

661 Descriptions for Part Properties of `Motor`:

662   • `Composition`

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

664   • `Component`

665   logical or physical entity that provides a capability.

666   4.3.32   Oil

667 leaf `Component` composed of a viscous liquid.

668 **4.3.32.1   Part Properties of Oil**

669 *Table 35* lists the Part Properties of `Oil`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 35: Part Properties of Oil

670 Descriptions for Part Properties of `Oil`:

671 • `Composition`

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

673 • `Component`

674   logical or physical entity that provides a capability.

4.3.33   Part

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

677 **4.3.33.1   PartOccurrence**

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

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

681 **4.3.33.2   Commonly Observed DataItem Types for PartOccurrence**

682 *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

4.3.33.2.1   FeatureOccurrence

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

4.3.33.3   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`

691   tabular representation of assessing elements of a *feature*.

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

694   • `MeasurementType`

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

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

697   • `CharacteristicPersistentId`

698   UUID of the *characteristic*.

699   • `CharacteristicStatus`

700   pass/fail result of the measurement.

701   • `Uncertainty`

702   *uncertainty* specified by `UncertaintyType`.

703   • `UncertaintyType`

704   method used to compute *standard uncertainty*.

705   • `MeasurementUnits`

706   engineering units of the measurement.

707   • `MeasurementValue`

708   measurement based on the measurement type.

709   4.3.34   Path

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

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

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

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

720   **4.3.34.1   Part Properties of Path**

721   *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

722   Descriptions for Part Properties of `Path`:

723   • `Program`

724   name of the logic or motion program being executed by the `Controller` compo-
725   nent.

726   • `PathFeedrateOverride.Programmed`

727   directive value without offsets and adjustments.

728   • `PathFeedrateOverride.Rapid`

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

730   • `RotaryVelocityOverride`

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

734   • `PathFeedrate`

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

736   • `PartCount`

737   aggregate count of parts.

738   **4.3.34.2   Commonly Observed DataItem Types for Path**

739   *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

4.3.35   Pot

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

4.3.35.1   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.

4.3.36   `<<deprecated>>`Power

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

4.3.37   PowerSupply

leaf `Component` that provides power to electric mechanisms.

755 **4.3.37.1 Part Properties of PowerSupply**

756 *Table 41* lists the Part Properties of `PowerSupply`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 41: Part Properties of PowerSupply

757 Descriptions for Part Properties of `PowerSupply`:

758 • `Composition`

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

760 • `Component`

761 logical or physical entity that provides a capability.

762 4.3.38 Process

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

765 **4.3.38.1 ProcessOccurrence**

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

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

769 **4.3.38.2 Commonly Observed DataItem Types for ProcessOccurrence**

770 *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

771 4.3.39   Pulley

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

774 4.3.39.1   Part Properties of Pulley

775 *Table 43* lists the Part Properties of `Pulley`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 43: Part Properties of Pulley

776 Descriptions for Part Properties of `Pulley`:

777 • `Composition`

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

779 • `Component`

780 logical or physical entity that provides a capability.

781 4.3.40   Pump

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

784 **4.3.40.1 Part Properties of Pump**

785 *Table 44* lists the Part Properties of `Pump`.

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

Table 44: Part Properties of Pump

786 Descriptions for Part Properties of `Pump`:

787 • `Composition`

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

789 • `Component`

790   logical or physical entity that provides a capability.

791 **4.3.41 Reel**

792 leaf `Component` composed of a rotary storage unit for material.

793 **4.3.41.1 Part Properties of Reel**

794 *Table 45* lists the Part Properties of `Reel`.

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

Table 45: Part Properties of Reel

795 Descriptions for Part Properties of `Reel`:

796 • `Composition`

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

798 • `Component`

799   logical or physical entity that provides a capability.

800 4.3.42 RemovalPot

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

803 4.3.42.1 Part Properties of RemovalPot

804 *Table 46* lists the Part Properties of `RemovalPot`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 46: Part Properties of RemovalPot

805 Descriptions for Part Properties of `RemovalPot`:

806 • `Composition`
807   functional part of a piece of equipment contained within a `Component`.

808 • `Component`
809   logical or physical entity that provides a capability.

810 4.3.43 Resource

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

813 4.3.43.1 Material

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

816 4.3.43.1.1 Stock

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

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

822 **4.3.43.2   Part Properties of Stock**

823 *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

824 Descriptions for Part Properties of `Stock`:

825 • `Material`

826   identifier of a material used or consumed in the manufacturing process.

827 • `Length.Remaining`

828   remaining total length of an object.

829 • `Length.Standard`

830   standard or original length of an object.

831 **4.3.43.3   Personnel**

832 `Resource` composed of an individual or individuals who either control, support, or oth-
833 erwise interface with a piece of equipment.

834 **4.3.43.4   Part Properties of Personnel**

835 *Table 48* lists the Part Properties of `Personnel`.

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

Table 48: Part Properties of Personnel

836 Descriptions for Part Properties of `Personnel`:

837 • `User.Operator`

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

839 • `User.Maintenance`

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

842 ## 4.3.44 ReturnPot

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

845 ### 4.3.44.1 Part Properties of ReturnPot

846 *Table 49* lists the Part Properties of `ReturnPot`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 49: Part Properties of ReturnPot

847 Descriptions for Part Properties of `ReturnPot`:

848 • `Composition`

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

850 • `Component`

851 logical or physical entity that provides a capability.

852 **4.3.45 SensingElement**

853 leaf `Component` that provides a signal or measured value.

854 **4.3.45.1 Part Properties of SensingElement**

855 *Table 50* lists the Part Properties of `SensingElement`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 50: Part Properties of SensingElement

856 Descriptions for Part Properties of `SensingElement`:

857 • `Composition`

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

859 • `Component`

860 logical or physical entity that provides a capability.

861 **4.3.46 Sensor**

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

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

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

869 See `SensorConfiguration` for more details on the use and configuration of a `Sen-`
870 `sor`.

871 **4.3.46.1 `<<deprecated>>`Thermostat**

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

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

875 **4.3.46.2 `<<deprecated>>`Vibration**

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

878 `Vibration` was **DEPRECATED** in *MTConnect Version 1.2* and was replaced by `Dis-`
879 `placement`, `Frequency` etc.

880 4.3.47   Spreader

881 leaf `Component` that flattens or spreading materials.

882 **4.3.47.1   Part Properties of Spreader**

883 *Table 51* lists the Part Properties of `Spreader`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 51: Part Properties of Spreader

884 Descriptions for Part Properties of `Spreader`:

885 • `Composition`

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

887 • `Component`

888   logical or physical entity that provides a capability.

889 4.3.48 StagingPot

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

892 4.3.48.1 Part Properties of StagingPot

893 *Table 52* lists the Part Properties of `StagingPot`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 52: Part Properties of StagingPot

894 Descriptions for Part Properties of `StagingPot`:

895   • `Composition`
896   functional part of a piece of equipment contained within a `Component`.

897   • `Component`
898   logical or physical entity that provides a capability.

899 4.3.49 Station

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

902 4.3.49.1 Part Properties of Station

903 *Table 53* lists the Part Properties of `Station`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 53: Part Properties of Station

904 Descriptions for Part Properties of `Station`:

905 • Composition

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

907 • Component

908   logical or physical entity that provides a capability.

909   4.3.50   StorageBattery

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

912   4.3.50.1   Part Properties of StorageBattery

913 *Table 54* lists the Part Properties of `StorageBattery`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 54: Part Properties of StorageBattery

914 Descriptions for Part Properties of `StorageBattery`:

915 • Composition

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

917 • Component

918   logical or physical entity that provides a capability.

919   4.3.51   Structure

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

921   4.3.51.1   Link

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

923 ## 4.3.52 Switch

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

925 ### 4.3.52.1 Part Properties of Switch

926 *Table 55* lists the Part Properties of `Switch`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 55: Part Properties of Switch

927 Descriptions for Part Properties of `Switch`:

928 • `Composition`

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

930 • `Component`

931 logical or physical entity that provides a capability.

932 ## 4.3.53 System

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

934 ### 4.3.53.1 Controller

935 `System` that provides regulation or management of a system or component. *Ref ISO*
936 *16484-5:2017*

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

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

941 **4.3.53.2 Part Properties of Controller**

942 *Table 56* 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 56: Part Properties of Controller

943 Descriptions for Part Properties of `Controller`:

944 • `Path`

945 `Component` that organizes an independent operation or function within a `Con-`
946 `troller`.

947 • `EmergencyStop`

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

950 • `System`

951 general purpose indication associated with an electronic component of a piece of
952 equipment or a controller that represents a fault that is not associated with the oper-
953 ator, program, or hardware.

954 • `ControllerMode`

955 current mode of the `Controller` component.

956 • `Communications`

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

958 • `LogicProgram`

959 indication that an error occurred in the logic program or programmable logic con-
960 troller (PLC) associated with a piece of equipment.

961 **4.3.53.3 Coolant**

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

964 **4.3.53.4 Part Properties of Coolant**

965 *Table 57* lists the Part Properties of `Coolant`.

Part Property name	Multiplicity
observesConcentration	0..1

Table 57: Part Properties of Coolant

966 Descriptions for Part Properties of `Coolant`:

967 • `Concentration`
968   percentage of one component within a mixture of components.

969 **4.3.53.5 Dielectric**

970 `System` that manages a chemical mixture used in a manufacturing process being per-
971 formed at that piece of equipment.

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

974 **4.3.53.6 Electric**

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

977 The electric system will provide all the data with regard to electric current, voltage, fre-
978 quency, etc. that applies to the piece of equipment as a functional unit. Data regarding
979 electric power that is specific to a `Component` will be reported for that specific `Compo-`
980 `nent`.

981 **4.3.53.7 Part Properties of Electric**

982 *Table 58* lists the Part Properties of `Electric`.

Part Property name	Multiplicity
observesWattage	0..1

Table 58: Part Properties of Electric

983 Descriptions for Part Properties of `Electric`:

984 • `Wattage`

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

986 **4.3.53.8 Enclosure**

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

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

992 **4.3.53.9 EndEffector**

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

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

995 **4.3.53.10 Feeder**

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

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

1000 **4.3.53.11 Hydraulic**

1001 `System` that provides movement and distribution of pressurized liquid throughout the
1002 piece of equipment.

1003 **4.3.53.12 Part Properties of Hydraulic**

1004 *Table 59* lists the Part Properties of `Hydraulic`.

Part Property name	Multiplicity
observesPressure	0..1

Table 59: Part Properties of Hydraulic

1005 Descriptions for Part Properties of `Hydraulic`:

1006 • `Pressure`

1007   force per unit area measured relative to atmospheric pressure.

1008   Commonly referred to as gauge pressure.

1009 **4.3.53.13 Lubrication**

1010 `System` that provides distribution and management of fluids used to lubricate portions of
1011 the piece of equipment.

1012 **4.3.53.14 Pneumatic**

1013 `System` that uses compressed gasses to actuate components or do work within the piece
1014 of equipment.

1015   Note: Actuation is usually performed using a cylinder.

1016 **4.3.53.15 Part Properties of Pneumatic**

1017 *Table 60* lists the Part Properties of `Pneumatic`.

Part Property name	Multiplicity
observesPressure	0..1

Table 60: Part Properties of Pneumatic

1018 Descriptions for Part Properties of `Pneumatic`:

1019 • `Pressure`

1020   force per unit area measured relative to atmospheric pressure.

1021   Commonly referred to as gauge pressure.

1022 4.3.53.16 ProcessPower

1023 `System` composed of a power source associated with a piece of equipment that supplies
1024 energy to the manufacturing process separate from the `Electric` system.

1025 For example, this could be the power source for an EDM machining process, an electro-
1026 plating line, or a welding system.

1027 4.3.53.17 Protective

1028 `System` that provides functions used to detect or prevent harm or damage to equipment
1029 or personnel.

1030 `Protective` does not include the information relating to the `Enclosure`.

1031 4.3.53.18 Heating

1032 `System` that delivers controlled amounts of heat to achieve a target temperature at a spec-
1033 ified heating rate.

1034   Note: As an example, Energy Delivery Method can be either through Electric
1035   heaters or Gas burners.

1036 4.3.53.19 Vacuum

1037 `System` that evacuates gases and liquids from an enclosed and sealed space to a controlled
1038 negative pressure or a molecular density below the prevailing atmospheric level.

1039 4.3.53.20 Cooling

1040 `System` that extracts controlled amounts of heat to achieve a target temperature at a spec-
1041 ified cooling rate.

1042   Note: As an example, Energy Extraction Method can be via cooling water
1043   pipes running through the chamber.

4.3.53.21 Pressure

1044

1045 `System` that delivers compressed gas or fluid and controls the pressure and rate of pres-
1046 sure change to a desired target set-point.

1047   Note: For example, Delivery Method can be a Compressed Air or N2 tank
1048   that is piped via an inlet valve to the chamber.

4.3.53.22 WorkEnvelope

1049

1050 `System` composed of the physical process execution space within a piece of equipment.

1051 `WorkEnvelope` **MAY** provide information regarding the physical workspace and the
1052 conditions within that workspace.

4.3.54 Table

1053

1054 leaf `Component` composed of a surface for holding an object or material.

4.3.54.1 Part Properties of Table

1055

1056 *Table 61* lists the Part Properties of `Table`.

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

Table 61: Part Properties of Table

1057 Descriptions for Part Properties of `Table`:

1058   • `Composition`
1059   functional part of a piece of equipment contained within a `Component`.

1060   • `Component`
1061   logical or physical entity that provides a capability.

1062 ## 4.3.55 Tank

1063 leaf `Component` composed of a receptacle or container that holds material.

1064 ### 4.3.55.1 Part Properties of Tank

1065 *Table 62* lists the Part Properties of `Tank`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 62: Part Properties of Tank

1066 Descriptions for Part Properties of `Tank`:

1067   • `Composition`

1068   functional part of a piece of equipment contained within a `Component`.

1069   • `Component`

1070   logical or physical entity that provides a capability.

1071 ## 4.3.56 Tensioner

1072 leaf `Component` that provides or applies a stretch or strain to another mechanism.

1073 ### 4.3.56.1 Part Properties of Tensioner

1074 *Table 63* lists the Part Properties of `Tensioner`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 63: Part Properties of Tensioner

1075 Descriptions for Part Properties of `Tensioner`:

1076   • `Composition`

1077   functional part of a piece of equipment contained within a `Component`.

1078   • `Component`

1079   logical or physical entity that provides a capability.

1080   4.3.57   TransferArm

1081   leaf `Component` that physically moves a tool from one location to another.

1082   4.3.57.1   Part Properties of TransferArm

1083   *Table 64* lists the Part Properties of `TransferArm`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 64: Part Properties of TransferArm

1084   Descriptions for Part Properties of `TransferArm`:

1085   • `Composition`

1086   functional part of a piece of equipment contained within a `Component`.

1087   • `Component`

1088   logical or physical entity that provides a capability.

1089   4.3.58   TransferPot

1090   leaf `Component` that is a `Pot` for a tool that is awaiting transfer from a `ToolMagazine`
1091   to *spindle* or `Turret`.

1092   4.3.58.1   Part Properties of TransferPot

1093   *Table 65* lists the Part Properties of `TransferPot`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 65: Part Properties of TransferPot

1094 Descriptions for Part Properties of `TransferPot`:

1095 • `Composition`

1096 functional part of a piece of equipment contained within a `Component`.

1097 • `Component`

1098 logical or physical entity that provides a capability.

4.3.59 Transformer

1100 leaf `Component` that transforms electric energy from a source to a secondary circuit.

4.3.59.1 Part Properties of Transformer

1102 *Table 66* lists the Part Properties of `Transformer`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 66: Part Properties of Transformer

1103 Descriptions for Part Properties of `Transformer`:

1104 • `Composition`

1105 functional part of a piece of equipment contained within a `Component`.

1106 • `Component`

1107 logical or physical entity that provides a capability.

1108 4.3.60 Valve

1109 leaf `Component` that halts or controls the flow of a liquid, gas, or other material through
1110 a passage, pipe, inlet, or outlet.

1111 4.3.60.1 Part Properties of Valve

1112 *Table 67* lists the Part Properties of `Valve`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 67: Part Properties of Valve

1113 Descriptions for Part Properties of `Valve`:

1114 • `Composition`
1115   functional part of a piece of equipment contained within a `Component`.

1116 • `Component`
1117   logical or physical entity that provides a capability.

1118 4.3.61 Vat

1119 leaf `Component` composed of a container that holds liquid or powdered materials.

1120 4.3.61.1 Part Properties of Vat

1121 *Table 68* lists the Part Properties of `Vat`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 68: Part Properties of Vat

1122 Descriptions for Part Properties of `Vat`:

1123 • Composition

1124   functional part of a piece of equipment contained within a Component.

1125 • Component

1126   logical or physical entity that provides a capability.

1127 4.3.62 Water

1128 leaf Component composed of H_2O.

1129 4.3.62.1 Part Properties of Water

1130 *Table 69* lists the Part Properties of Water.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 69: Part Properties of Water

1131 Descriptions for Part Properties of Water:

1132 • Composition

1133   functional part of a piece of equipment contained within a Component.

1134 • Component

1135   logical or physical entity that provides a capability.

1136 4.3.63 Wire

1137 leaf Component composed of a string like piece or filament of relatively rigid or flexible
1138 material provided in a variety of diameters.

1139 4.3.63.1 Part Properties of Wire

1140 *Table 70* lists the Part Properties of Wire.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 70: Part Properties of Wire

1141 Descriptions for Part Properties of `Wire`:

1142 • `Composition`

1143 functional part of a piece of equipment contained within a `Component`.

1144 • `Component`

1145 logical or physical entity that provides a capability.

1146 **4.3.64 Workpiece**

1147 leaf `Component` composed of an object or material on which a form of work is per-
1148 formed.

1149 **4.3.64.1 Part Properties of Workpiece**

1150 *Table 71* lists the Part Properties of `Workpiece`.

Part Property name	Multiplicity
hasComposition	0
hasComponent	0

Table 71: Part Properties of Workpiece

1151 Descriptions for Part Properties of `Workpiece`:

1152 • `Composition`

1153 functional part of a piece of equipment contained within a `Component`.

1154 • `Component`

1155 logical or physical entity that provides a capability.

1156   4.4   Component Organizer Types

1157 This section provides semantic information for the types of `Component` that are used to
1158 *organize* other `Component` types.

1159   4.4.1   `<<organizer>>`Adapters

1160 `Component` that *organizes* `Adapter` types.

1161   4.4.2   `<<organizer>>`Auxiliaries

1162 `Component` that *organizes* `Auxiliary` types.

1163   4.4.3   `<<organizer>>`Axes

1164 `Component` that *organizes* `Axis` types.

1165   4.4.4   `<<organizer>>`Controllers

1166 `Component` that *organizes* `Controller` entities.

1167   4.4.5   `<<organizer>>`Interfaces

1168 `Component` that *organizes* `Interface` types.

1169   4.4.6   `<<organizer>>`Parts

1170 `Component` that *organizes* `Part` types.

1171 **4.4.7 `<<organizer>>`Processes**

1172 Component that *organizes* `Process` types.

1173 **4.4.8 `<<organizer>>`Resources**

1174 Component that *organizes* `Resource` types.

1175 **4.4.8.1 `<<organizer>>`Materials**

1176 `Resources` that *organizes* `Material` types.

1177 **4.4.9 `<<organizer>>`Structures**

1178 Component that *organizes* `Structure` types.

1179 **4.4.10 `<<organizer>>`Systems**

1180 Component that *organizes* `System` types.

5 Compositions Model

1181

1182 `Composition` entities are used to describe the lowest level physical building blocks of
1183 a piece of equipment contained within a `Component`. By referencing a specific `Com-`
1184 `position` entity, further clarification and meaning to data associated with a specific
1185 `Component` can be achieved.

1186 Both `Component` and `Composition` entities are *lower level* entities representing the
1187 sub-parts of the parent `Component`. However, there are distinct differences between
1188 `Component` and `Composition` type entities.

1189 `Component` entities may be further defined with *lower level* `Component` entities and
1190 may have associated *DataItems*.

1191 `Composition` entities represent the lowest level physical part of a piece of equipment.
1192 They **MUST NOT** be further defined with *lower level* `Component` entities and they
1193 **MUST NOT** have *DataItems* directly associated with them. They do provide additional
1194 information that can be used to enhance the specificity of *DataItems* associated with the
1195 parent `Component`.

5.1 Compositions

1196

1197 This section provides semantic information for the `Composition` entity.

1198 See Figure 2 for the `Composition` model diagram.

1199   Note: See *Example 3* for the XML representation of the same example.

5.1.1 Composition

1200

1201 functional part of a piece of equipment contained within a `Component`.

1202 `Composition` **MUST NOT** have child `Component`, `Composition`, or `DataItems`
1203 elements.

5.1.1.1 Value Properties of Composition

1204

1205 *Table 72* lists the Value Properties of `Composition`.
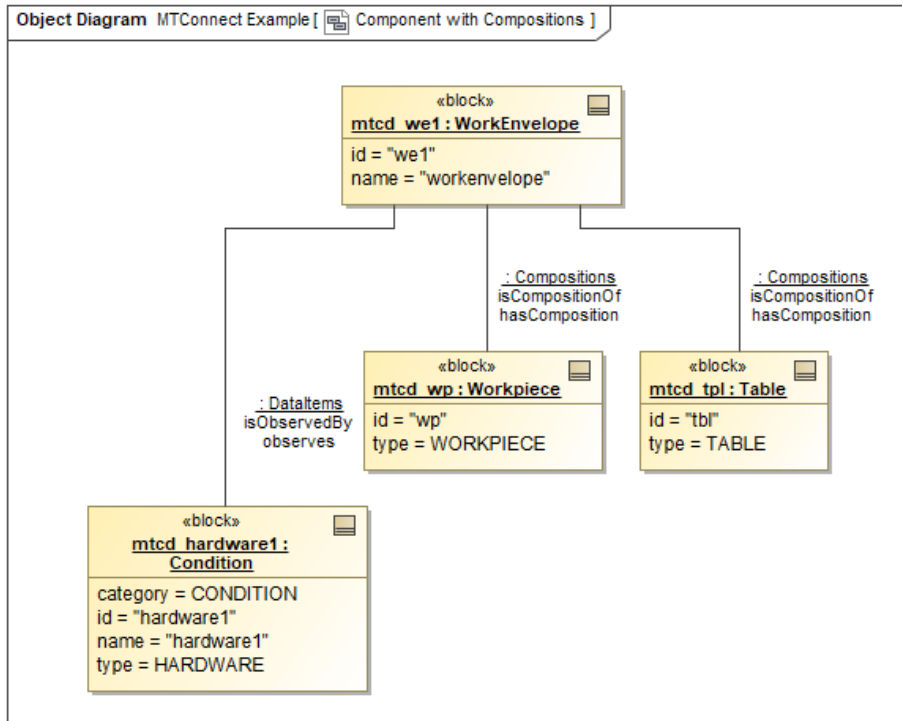
Figure 4: Component with Compositions Example

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

Table 72: Value Properties of Composition

1206   Descriptions for Value Properties of `Composition`:

1207   • `id`

1208   unique identifier for the `Composition` element.

1209   • `type`

1210   type of `Composition`.

1211   The value of `type` **MUST** be one of the `CompositionTypeEnum` enumeration.

1212   • `name`

1213   name of the `Composition` element.

1214 • uuid

1215 universally unique identifier for the `Composition`.

5.1.1.2 Part Properties of Composition

1217 *Table 73* lists the Part Properties of `Composition`.

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

Table 73: Part Properties of Composition

1218 Descriptions for Part Properties of `Composition`:

1219 • Description

1220 descriptive content.

1221 See *Section 4.1.2 - Description*.

1222 • Configuration

1223 technical information about an entity describing its physical layout, functional char-
1224 acteristics, and relationships with other entities.

1225 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`.
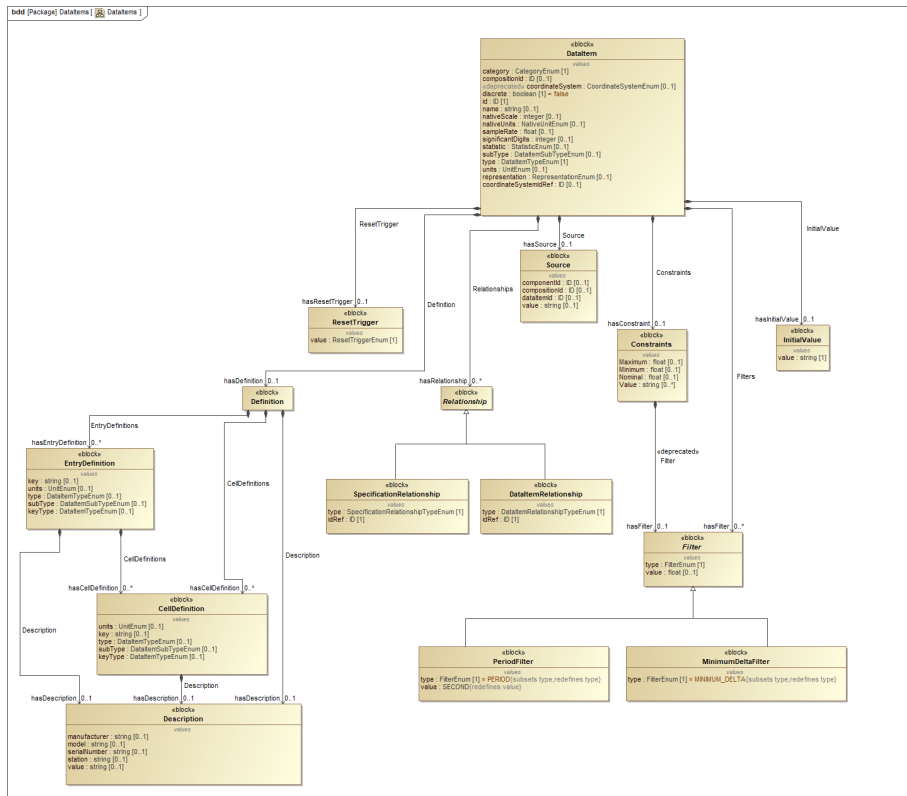
Figure 5: DataItems

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

1248 Descriptions for Value Properties of `DataItem`:

1249 • `category`

1250 specifies the kind of information provided by a data item.

1251 `CategoryEnum` Enumeration:

1252 – `CONDITION`

1253 information about the health of a piece of equipment and its ability to function.

1254 – `EVENT`

1255 discrete piece of information from the piece of equipment.

1256 – `SAMPLE`

1257 continuously variable or analog data value.

1258 A continuous value can be measured at any point-in-time and will always pro-
1259 duce a result.

1260 • `compositionId`

1261 identifier attribute of the `Composition` that the reported data is most closely as-
1262 sociated.

1263 • `<<deprecated>> coordinateSystem`

1264 for measured values relative to a coordinate system like `Position`, the coordinate
1265 system used may be reported.

1266 **DEPRECATED** in *Version 2.0*. Replaced by `coordinateSystemIdRef`.

1267 • `discrete`

1268 indication signifying whether each value reported for the *Observation* is significant
1269 and whether duplicate values are to be suppressed.

1270 If a value is not defined for `discrete`, the default value **MUST** be `false`.

1271 • `id`

1272 unique identifier for this data item.

1273 • `name`

1274 name of the data item.

1275 • `nativeScale`

1276 used to convert the reported value to represent the original measured value.

1277 • nativeUnits

1278   native units of measurement for the reported value of the data item.

1279   <<extensible>> NativeUnitEnum Enumeration:

1280   – AMPERE_HOUR

1281   electric charge in ampere hour.

1282   – BAR

1283   pressure in Bar.

1284   – CENTIPOISE

1285   measure of viscosity.

1286   – DEGREE/MINUTE

1287   rotational velocity in degrees per minute.

1288   – FAHRENHEIT

1289   temperature in Fahrenheit.

1290   – FOOT

1291   feet.

1292   – FOOT/MINUTE

1293   feet per minute.

1294   – FOOT/SECOND

1295   feet per second.

1296   – FOOT/SECOND^2

1297   acceleration in feet per second squared.

1298   – FOOT_3D

1299   point in space identified by X, Y, and Z positions and represented by a space-
1300   delimited set of numbers each expressed in feet.

1301   – GALLON/MINUTE

1302   gallons per minute.

1303   – GRAVITATIONAL_ACCELERATION

1304   acceleration relative to earth's gravity given in METER/SECOND^2.

1305   Note 1 to entry: At different points on Earth's surface, the free-fall
1306   acceleration ranges from 9.764 to 9.834 m/s2 (Wikipedia: Gravita-
1307   tional Acceleration). The constant can be customized depending on
1308   the location in the universe.

1309   Note 2 to entry: In the standard, it is assumed that Earth's average
1310   value of gravitational acceleration is 9.90665 m/s2.

1311   – GRAVITATIONAL_FORCE

1312   *MASS* × *GRAVITATIONAL_ACCELERATION* (g) given in METER/SECOND^2.

1313   – HOUR

1314   measurement of time in hours.

1315   – INCH

1316   inches.

1317   – INCH/MINUTE

1318   inches per minute.

1319   – INCH/SECOND

1320   inches per second.

1321   – INCH/SECOND^2

1322   acceleration in inches per second squared.

1323   – INCH_3D

1324   point in space identified by X, Y, and Z positions and represented by a space-
1325   delimited set of numbers each expressed in inches.

1326   – INCH_POUND

1327   measure of torque in inch pounds.

1328   – KELVIN

1329   measurement of temperature.

1330   – KILOWATT

1331   measurement in kilowatt.

1332   – KILOWATT_HOUR

1333   kilowatt hours which is 3.6 mega joules.

1334   – LITER/MINUTE

1335   measurement of rate of flow of a fluid.

1336   – MILLIMETER/MINUTE

1337   velocity in millimeters per minute.

1338   – MILLIMETER_MERCURY

1339   pressure in Millimeter of Mercury (mmHg).

1340   – MINUTE

1341   measurement of time in minutes.

1342   – OTHER

1343   unsupported unit.

1344   – PASCAL/MINUTE

1345   pascal per minute.

1346   – POUND
1347   US pounds.

1348   – POUND/INCH^2
1349   pressure in pounds per square inch (PSI).

1350   – RADIAN
1351   angle in radians.

1352   – RADIAN/MINUTE
1353   velocity in radians per minute.

1354   – RADIAN/SECOND
1355   rotational acceleration in radian per second squared.

1356   – RADIAN/SECOND^2
1357   rotational acceleration in radian per second squared.

1358   – TORR
1359   pressure in Torr.

1360   • sampleRate
1361   rate at which successive samples of a data item are recorded by a piece of equipment.

1362   • significantDigits
1363   number of significant digits in the reported value.

1364   • statistic
1365   type of statistical calculation performed on a series of data samples to provide the
1366   reported data value.

1367   StatisticEnum Enumeration:

1368   – AVERAGE
1369   mathematical average value calculated for the data item during the calculation
1370   period.

1371   – <<deprecated>> KURTOSIS
1372   **DEPRECATED** in *Version 1.6*. ∼∼A measure of the "peakedness" of a prob-
1373   ability distribution; i.e., the shape of the distribution curve.∼∼

1374   – MAXIMUM
1375   maximum or peak value recorded for the data item during the calculation pe-
1376   riod.

1377   – MEDIAN
1378   middle number of a series of numbers.

1379   – MINIMUM

1380   minimum value recorded for the data item during the calculation period.

1381   – MODE

1382   number in a series of numbers that occurs most often.

1383   – RANGE

1384   difference between the maximum and minimum value of a data item during the

1385   calculation period. Also represents Peak-to-Peak measurement in a waveform.

1386   – ROOT_MEAN_SQUARE

1387   mathematical Root Mean Square (RMS) value calculated for the data item dur-

1388   ing the calculation period.

1389   – STANDARD_DEVIATION

1390   statistical Standard Deviation value calculated for the data item during the cal-

1391   culation period.

1392   • subType

1393   sub-categorization of the data item type.

1394   <<extensible>> DataItemSubTypeEnum Enumeration:

1395   – ABORTED

1396   actions or activities that were attempted, but terminated before they could be

1397   completed.

1398   – ABSOLUTE

1399   relating to or derived in the simplest manner from the fundamental units or

1400   measurements.

1401   – ACTION

1402   indication of the operating state of a mechanism.

1403   – ACTIVE

1404   relating to logic or motion program currently executing.

1405   – ACTIVITY

1406   phase or segment of a recipe or program.

1407   – ACTUAL

1408   measured or reported value of an observation.

1409   – ALL

1410   all actions, items, or activities being counted independent of the outcome.

1411   – <<deprecated>> ALTERNATING

1412   measurement of alternating voltage or current. If not specified further in statis-

1413   tic, defaults to RMS voltage.

1414   **DEPRECATED** in *Version 1.6*.

1415 – AUXILIARY

1416 when multiple locations on a piece of bar stock being feed by a bar feeder are
1417 referenced as the indication of whether the end of that piece of bar stock has
1418 been reached.

1419 – A_SCALE

1420 A-Scale weighting factor on the frequency scale.

1421 – BAD

1422 actions, items, or activities being counted that do not conform to specification
1423 or expectation.

1424 – BATCH

1425 group of parts produced in a batch.

1426 – BINARY

1427 observed as a binary data type.

1428 – BOOLEAN

1429 observed as a boolean data type.

1430 – BRINELL

1431 scale to measure the resistance to deformation of a surface.

1432 – B_SCALE

1433 B-Scale weighting factor on the frequency scale.

1434 – COMMANDED

1435 directive value including adjustments such as an offset or overrides.

1436 – COMPLETE

1437 associated with the completion of an activity or event.

1438 – CONSUMED

1439 amount of material consumed from an object or container during a manufac-
1440 turing process.

1441 – CONTROL

1442 state of the enabling signal or control logic that enables or disables the function
1443 or operation of the entity.

1444 – C_SCALE

1445 C-Scale weighting factor on the frequency scale.

1446 – DELAY

1447 elapsed time of a temporary halt of action.

1448 – DETECT

1449 indicated by the presence or existence of something.

1450 – <<deprecated>> `DIRECT`

DC current or voltage.

DEPRECATED in *Version 1.6*.

1453 – `DRY_RUN`

setting or operator selection used to execute a test mode to confirm the execution of machine functions.

1456 – `D_SCALE`

D-Scale weighting factor on the frequency scale.

1458 – `ENDED`

boundary when an activity or an event terminates.

1460 – `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*

1463 – `EXPIRATION`

relating to the expiration or end of useful life for a material or other physical item.

1466 – `FAILED`

actions or activities that were attempted , but failed to complete or resulted in an unexpected or unacceptable outcome.

1469 – `FIRST_USE`

relating to the first use of a material or other physical item.

1471 – `GATEWAY`

Gateway for the `Component` network.

1473 – `GOOD`

actions, items, or activities being counted that conform to specification or expectation.

1476 – `HEAT_TREAT`

material heat number.

1478 – `INCREMENTAL`

relating to or derived from the last observation.

1480 – `INSTALL_DATE`

date the hardware or software was installed.

1482 – `IPV4_ADDRESS`

IPV4 network address of the `Component`.

1484 – `IPV6_ADDRESS`

IPV6 network address of the `Component`.

1486   – ISO_STEP_EXECUTABLE

1487   reference to a ISO 10303 Executable.

1488   – JOG

1489   relating to momentary activation of a function or a movement.

1490   **DEPRECATION WARNING**: May be deprecated in the future.

1491   – LATERAL

1492   indication of the position of a mechanism that may move in a lateral direction.

1493   – LEEB

1494   scale to measure the elasticity of a surface.

1495   – LENGTH

1496   reference to a length type tool offset variable.

1497   – LICENSE

1498   license code to validate or activate the hardware or software.

1499   – LINE

1500   state of the power source.

1501   – LINEAR

1502   direction of motion of a linear motion.

1503   – LOADED

1504   indication that the subparts of a piece of equipment are under load.

1505   – LOT

1506   group of parts tracked as a lot.

1507   – MACHINE_AXIS_LOCK

1508   setting or operator selection that changes the behavior of the controller on a
1509   piece of equipment.

1510   – MAC_ADDRESS

1511   Media Access Control Address. The unique physical address of the network
1512   hardware.

1513   – MAIN

1514   relating to the primary logic or motion program currently being executed.

1515   – MAINTENANCE

1516   relating to maintenance on the piece of equipment.

1517   – MANUAL_UNCLAMP

1518   indication of the state of an operator controlled interlock that can inhibit the
1519   ability to initiate an unclamp action of an electronically controlled chuck.

1520 – MANUFACTURE

1521 related to the production of a material or other physical item.

1522 – MANUFACTURER

1523 corporate identity for the maker of the hardware or software.

1524 – MAXIMUM

1525 maximum value.

1526 – MINIMUM

1527 minimum value.

1528 – MOHS

1529 scale to measure the resistance to scratching of a surface.

1530 – MOTION

1531 indication of the open or closed state of a mechanism.

1532 – NO_SCALE

1533 no weighting factor on the frequency scale.

1534 – OPERATING

1535 piece of equipment that is powered or performing any activity.

1536 – OPERATION

1537 step of a discrete manufacturing process.

1538 – OPERATOR

1539 relating to the person currently responsible for operating the piece of equip-
1540 ment.

1541 – OPTIONAL_STOP

1542 setting or operator selection that changes the behavior of the controller on a
1543 piece of equipment.

1544 – ORDER_NUMBER

1545 authorization of a process occurrence.

1546 – OVERRIDE

1547 overridden value.

1548 – PART

1549 amount included in the *part*.

1550 – PART_FAMILY

1551 group of parts having similarities in geometry, manufacturing process, and/or
1552 functions.

1553 – PART_NAME

1554 word or set of words by which a part is known, addressed, or referred to.

1555 – PART_NUMBER

1556 particular part design or model.

1557 – POWERED

1558 piece of equipment is powered and functioning or `Component` that are re-
1559 quired to remain on are powered.

1560 – PRIMARY

1561 main or most important location of a piece of bar stock.

1562 – PROBE

1563 position provided by a measurement probe.

1564 **DEPRECATION WARNING**: May be deprecated in the future.

1565 – PROCESS

1566 relating to production of a part or product on a piece of equipment.

1567 – PROCESS_NAME

1568 word or set of words by which a process being executed (process occurrence)
1569 by the device is known, addressed, or referred to.

1570 – PROCESS_PLAN

1571 process plan that a process occurrence belongs to.

1572 – PROCESS_STEP

1573 step in the process plan that this occurrence corresponds to.

1574 – PROGRAMMED

1575 directive value without offsets and adjustments.

1576 – RADIAL

1577 reference to a radial type tool offset variable.

1578 – RAPID

1579 performing an operation faster or in less time than nominal rate.

1580 – RAW_MATERIAL

1581 material that is used to produce parts.

1582 – RECIPE

1583 process as part of product production; can be a subprocess of a larger process.

1584 – RELEASE_DATE

1585 date the hardware or software was released for general use.

1586 – REMAINING

1587 remaining measure or count of an action, object or activity.

1588 – REQUEST

1589 *request* by an `Interface` for a task.

1590   – RESPONSE

1591   *response* by an `Interface` to a *request* for a task.

1592   – ROCKWELL

1593   scale to measure the resistance to deformation of a surface.

1594   – ROTARY

1595   direction of a rotary motion using the right hand rule convention.

1596   – SCHEDULE

1597   identity of a control program that is used to specify the order of execution of
1598   other programs.

1599   – SEGMENT

1600   phase of a recipe process.

1601   – SERIAL_NUMBER

1602   serial number that uniquely identifies a specific part.

1603   – SET_UP

1604   relating to the preparation of a piece of equipment for production or restoring
1605   the piece of equipment to a neutral state after production.

1606   – SHORE

1607   scale to measure the resistance to deformation of a surface.

1608   – SINGLE_BLOCK

1609   setting or operator selection that changes the behavior of the controller on a
1610   piece of equipment.

1611   – STANDARD

1612   standard measure of an object or an action.

1613   – START

1614   boundary when an activity or an event commences.

1615   – SUBNET_MASK

1616   SubNet mask for the `Component` network.

1617   – SWITCHED

1618   indication of the activation state of a mechanism represented by a `Composi-`
1619   `tion`.

1620   – TARGET

1621   goal of the operation or process.

1622   – TARGET_COMPLETION

1623   relating to the end or completion of an activity or event.

1624 — TOOL_CHANGE_STOP

1625 setting or operator selection that changes the behavior of the controller on a
1626 piece of equipment.

1627 — USEABLE

1628 remaining usable measure of an object or action.

1629 — UUID

1630 universally unique identifier as specified in ISO 11578 or RFC 4122.

1631 — VERSION

1632 version of the hardware or software.

1633 — VERTICAL

1634 indication of the position of a mechanism that may move in a vertical direction.

1635 — VICKERS

1636 scale to measure the resistance to deformation of a surface.

1637 — VLAN_ID

1638 layer2 Virtual Local Network (VLAN) ID for the `Component` network.

1639 — WASTE

1640 amount discarded.

1641 — WIRELESS

1642 identifies whether the connection type is wireless.

1643 — WORKING

1644 piece of equipment performing any activity, the equipment is active and per-
1645 forming a function under load or not.

1646 • type

1647 type of data being measured. See *Section 6.5 - DataItem Types*.

1648 The value of type **MUST** be one of the `DataItemTypeEnum` enumeration.

1649 • units

1650 unit of measurement for the reported value of the data item.

1651 <<extensible>> UnitEnum Enumeration:

1652 — AMPERE
1653 amps.

1654 — CELSIUS
1655 degrees Celsius.

1656   – COULOMB

1657   electric charge in coulombs (C).

1658   – COUNT

1659   count of something.

1660   – COUNT/SECOND

1661   counts per second.

1662   – CUBIC_MILLIMETER

1663   geometric volume in millimeters.

1664   – CUBIC_MILLIMETER/SECOND

1665   change of geometric volume per second.

1666   – CUBIC_MILLIMETER/SECOND^2

1667   change in geometric volume per second squared.

1668   – DECIBEL

1669   sound level.

1670   – DEGREE

1671   angle in degrees.

1672   – DEGREE/SECOND

1673   angular degrees per second.

1674   – DEGREE/SECOND^2

1675   angular acceleration in degrees per second squared.

1676   – DEGREE_3D

1677   space-delimited, floating-point representation of the angular rotation in degrees
1678   around the X, Y, and Z axes relative to a cartesian coordinate system respec-
1679   tively in order as A, B, and C.

1680   If any of the rotations is not known, it **MUST** be zero (0).

1681   – GRAM

1682   gram.

1683   – GRAM/CUBIC_METER

1684   gram per cubic meter.

1685   – HERTZ

1686   frequency measured in cycles per second.

1687   – JOULE

1688   measurement of energy.

1689   – KILOGRAM

1690   kilograms.

1691   − LITER
1692   measurement of volume of a fluid.

1693   − LITER/SECOND
1694   liters per second.

1695   − METER/SECOND^2
1696   acceleration in meters per second squared.

1697   − MICRO_RADIAN
1698   measurement of tilt.

1699   − MILLIGRAM
1700   milligram.

1701   − MILLIGRAM/CUBIC_MILLIMETER
1702   milligram per cubic millimeter.

1703   − MILLILITER
1704   milliliter.

1705   − MILLIMETER
1706   millimeters.

1707   − MILLIMETER/REVOLUTION
1708   millimeters per revolution.

1709   − MILLIMETER/SECOND
1710   millimeters per second.

1711   − MILLIMETER/SECOND^2
1712   acceleration in millimeters per second squared.

1713   − MILLIMETER_3D
1714   point in space identified by X, Y, and Z positions and represented by a space-
1715   delimited set of numbers each expressed in millimeters.

1716   − NEWTON
1717   force in Newtons.

1718   − NEWTON_METER
1719   torque, a unit for force times distance.

1720   − OHM
1721   measure of electrical resistance.

1722   − PASCAL
1723   pressure in Newtons per square meter.

1724   − PASCAL/SECOND
1725   pascal per second.

1726   – PASCAL_SECOND

1727   measurement of viscosity.

1728   – PERCENT

1729   percentage.

1730   – PH

1731   measure of the acidity or alkalinity of a solution.

1732   – REVOLUTION/MINUTE

1733   revolutions per minute.

1734   – REVOLUTION/SECOND

1735   rotational velocity in revolution per second.

1736   – REVOLUTION/SECOND^2

1737   revolutions per second squared.

1738   – SECOND

1739   measurement of time.

1740   – SIEMENS/METER

1741   measurement of electrical conductivity.

1742   – UNIT_VECTOR_3D

1743   3D Unit Vector.

1744   Space delimited list of three floating point numbers.

1745   – VOLT

1746   volts.

1747   – VOLT_AMPERE

1748   measurement of the apparent power in an electrical circuit, equal to the product
1749   of root-mean-square (RMS) voltage and RMS current (commonly referred to
1750   as VA).

1751   – VOLT_AMPERE_REACTIVE

1752   measurement of reactive power in an AC electrical circuit (commonly referred
1753   to as VAR).

1754   – WATT

1755   watts.

1756   – WATT_SECOND

1757   measurement of electrical energy, equal to one Joule.

1758   • representation

1759   description of a means to interpret data consisting of multiple data points or samples
1760   reported as a single value.

1761   If `representation` is not specified, it **MUST** be determined to be `VALUE`.

1762   `RepresentationEnum` Enumeration:

1763   – `DATA_SET`
1764   reported value(s) are represented as a set of *key-value pairs*.
1765   Each reported value in the *data set* **MUST** have a unique key.

1766   – `<<deprecated>> DISCRETE`
1767   **DEPRECATED** as a `representation` in *MTConnect Version 1.5*. Re-
1768   placed by the `discrete,DataItem` attribute of a `DataItem`.

1769   – `TABLE`
1770   two dimensional set of *key-value pairs* where the `Entry` represents a row, and
1771   the value is a set of *key-value pair* `Cell` elements.
1772   A *table* follows the same behavior as the *data set* for change tracking, clearing,
1773   and history. When an `Entry` changes, all `Cell` elements update as a single
1774   unit following the behavior of a *data set*.

1775   Note: It is best to use the `Variable DataItem type` if the `Cell`
1776   elements represent multiple semantic types.

1777   Each `Entry` in the *table* **MUST** have a unique key. Each `Cell` of each `En-`
1778   `try` in the *table* **MUST** have a unique key.
1779   See `Representation` in *MTConnect Standard: Part 3.0 - Observation In-*
1780   *formation Model*, for a description of `Entry` and `Cell` elements.

1781   – `TIME_SERIES`
1782   series of sampled data.
1783   The data is reported for a specified number of samples and each sample is
1784   reported with a fixed period.

1785   – `VALUE`
1786   measured value of the sample data.
1787   If no `representation,DataItem` is specified for a data item, the `rep-`
1788   `resentation,DataItem` **MUST** be determined to be `VALUE`.

1789   • `coordinateSystemIdRef`
1790   associated `CoordinateSystem` context for the `DataItem`.

6.1.1.2   Reference Properties of DataItem

1792 *Table 75* lists the Reference Properties of `DataItem`.

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

Table 75: Reference Properties of DataItem

1793 Descriptions for Reference Properties of `DataItem`:

1794 • `Observation`

1795   abstract entity that provides telemetry data for a `DataItem` at a point in time.

1796   `Observations` groups one or more `Observation` entities made for the `DataItem`
1797   entity.

1798   `Component` observes `DataItem` entities to create `Observation` entities for
1799   the `DataItem` entities.

1800   See *MTConnect Standard: Part 3.0 - Observation Information Model* for the `Ob-`
1801   `servation` model.

1802   Note 1 to entry: In the XML representation, `Observation` entities
1803   **MUST NOT** appear in the *MTConnectDevices Response Document*.

1804   Note 2 to entry: In the XML representation, `Observation` entities
1805   **MUST** appear only in the *MTConnectStreams Response Document*.

1806 6.1.1.3 Part Properties of DataItem

1807 *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

1808 Descriptions for Part Properties of `DataItem`:

1809 • `Source`

1810 identifies the `Component`, `DataItem`, or `Composition` from which a mea-
1811 sured value originates.

1812 See *Section 6.2.2 - Source*.

1813 • `Constraints`

1814 *organizes* a set of expected values that can be reported for a `DataItem`.

1815 `Constraints` organizes a set of expected values that can be reported for the
1816 `DataItem`. See *Section 6.2.7 - Constraints*.

1817 • `Filter`

1818 provides a means to control when an *agent* records updated information for a `DataItem`.

1819 `Filters` groups one or more `Filter` entities associated with the `DataItem`.

1820 • `InitialValue`

1821 starting value for a `DataItem` as well as the value to be set for the `DataItem`
1822 after a reset event.

1823 • `ResetTrigger`

1824 type of event that may cause a reset to occur.

1825 • `Definition`

1826 defines the meaning of `Entry` and `Cell` elements associated with the `DataItem`
1827 when the `representation` is either `DATA` or `TABLE`.

1828 See *Section 6.2.8 - Definition*.

1829 • `AbstractDataItemRelationship`

1830 association between a `DataItem` and another entity.

1831 `Relationships` groups one or more `DataItemRelationship` and `Spec-`
1832 `ificationRelationship`. See *Section 8.5 - Relationships*.

1833 6.2 Properties of DataItem

1834 This section provides additional semantic information for the Part Properties of `DataItem`.

1835 Note: See *Section B.2 - DataItems Schema Diagrams* for XML schema of the
1836 elements for `DataItem`.

1837 6.2.1 ResetTrigger

1838 type of event that may cause a reset to occur.

1839 `ResetTriggerEnum` Enumeration:

1840 • `ACTION_COMPLETE`

1841 observation of the `DataItem` that is measuring an action or operation is to be reset
1842 upon completion of that action or operation.

1843 • `ANNUAL`

1844 observation of the `DataItem` is to be reset at the end of a 12-month period.

1845 • `DAY`

1846 observation of the `DataItem` is to be reset at the end of a 24-hour period.

1847 • `LIFE`

1848 observation of the `DataItem` is not reset and accumulates for the entire life of the
1849 piece of equipment.

1850 • `MAINTENANCE`

1851 observation of the `DataItem` is to be reset upon completion of a maintenance
1852 event.

1853 • `MONTH`

1854 observation of the `DataItem` is to be reset at the end of a monthly period.

1855 • `POWER_ON`

1856 observation of the `DataItem` is to be reset when power was applied to the piece of
1857 equipment after a planned or unplanned interruption of power has occurred.

1858 • `SHIFT`

1859 observation of the `DataItem` is to be reset at the end of a work shift.

1860 • `WEEK`

1861 observation of the `DataItem` is to be reset at the end of a 7-day period.

1862 ## 6.2.2   Source

1863 identifies the `Component`, `DataItem`, or `Composition` from which a measured value
1864 originates.

1865 The value of `Source` **MUST** be `string`.

1866 ### 6.2.2.1   Value Properties of Source

1867 *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

1868 Descriptions for Value Properties of `Source`:

1869 - `componentId`

1870 identifier of the `Component` that represents the physical part of a piece of equip-
1871 ment where the data represented by the `DataItem` originated.

1872 - `compositionId`

1873 identifier of the `Composition` that represents the physical part of a piece of equip-
1874 ment where the data represented by the `DataItem` originated.

1875 - `dataItemId`

1876 identifier of the `DataItem` that represents the originally measured value of the data
1877 referenced by this `DataItem`.

1878 ## 6.2.3   InitialValue

1879 starting value for a `DataItem` as well as the value to be set for the `DataItem` after a
1880 reset event.

1881 The value of `InitialValue` **MUST** be `string`.

1882 ## 6.2.4 Filter

1883 provides a means to control when an *agent* records updated information for a `DataItem`.

1884 The value of `Filter` **MUST** be `float`.

1885 ### 6.2.4.1 Value Properties of Filter

1886 *Table 78* lists the Value Properties of `Filter`.

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

Table 78: Value Properties of Filter

1887 Descriptions for Value Properties of `Filter`:

1888   • `type`

1889   type of `Filter`.

1890   `FilterEnum` Enumeration:

1891   – `MINIMUM_DELTA`

1892   new value **MUST NOT** be reported for a data item unless the measured value

1893   has changed from the last reported value by at least the delta given as the value

1894   of this element.

1895   The value of `Filter` **MUST** be an absolute value using the same units as the

1896   reported data.

1897   – `PERIOD`

1898   data reported for a data item is provided on a periodic basis. The `PERIOD` for

1899   reporting data is defined in the value of the `Filter`.

1900   The value of `Filter` **MUST** be an absolute value reported in seconds repre-

1901   senting the time between reported samples of the value of the data item.

1902 ## 6.2.5 MinimumDeltaFilter

1903 new value **MUST NOT** be reported for a data item unless the measured value has changed
1904 from the last reported value by at least the delta given as the value of this element.

1905 The value of `Filter` **MUST** be an absolute value using the same units as the reported
1906 data.

6.2.6 PeriodFilter

1908 data reported for a data item is provided on a periodic basis. The `PERIOD` for reporting
1909 data is defined in the value of the `Filter`.

1910 The value of `Filter` **MUST** be an absolute value reported in seconds representing the
1911 time between reported samples of the value of the data item.

1912 The value of `PeriodFilter` **MUST** be `SECOND`. See *Section 9.1.9 - SECOND*.

6.2.7 Constraints

1914 *organizes* a set of expected values that can be reported for a `DataItem`.

6.2.7.1 Value Properties of Constraints

1916 *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

1917 Descriptions for Value Properties of `Constraints`:

1918 • `Maximum`

1919   numeric upper constraint.

1920   If the data reported for a data item is a range of numeric values, the expected value
1921   reported **MAY** be described with an upper limit defined by this constraint.

1922 • `Minimum`

1923   numeric lower constraint.

1924   If the data reported for a data item is a range of numeric values, the expected value
1925   reported **MAY** be described with a lower limit defined by this constraint.

1926   • `Nominal`

1927   numeric target or expected value.

1928   • `Value`

1929   single data value that is expected to be reported for a `DataItem`.

1930   `Value` **MUST NOT** be used in conjunction with any other `Constraint` ele-
1931   ments.

1932 6.2.7.2   Part Properties of Constraints

1933 *Table 80* lists the Part Properties of `Constraints`.

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

Table 80: Part Properties of Constraints

1934 Descriptions for Part Properties of `Constraints`:

1935   • `Filter`

1936   provides a means to control when an *agent* records updated information for a `DataItem`.

1937   **DEPRECATED** in *MTConnect Version 1.4*. Moved to the `Filters`. See *Sec-*
1938   *tion 6.2 - Properties of DataItem*.

1939 6.2.8   Definition

1940 defines the meaning of `Entry` and `Cell` elements associated with the `DataItem` when
1941 the `representation` is either `DATA` or `TABLE`.

1942 6.2.8.1   Part Properties of Definition

1943 *Table 81* lists the Part Properties of `Definition`.

Part Property name	Multiplicity
CellDefinition (organized by CellDefinitions)	0..*
Description	0..1
EntryDefinition (organized by EntryDefinitions)	0..*

Table 81: Part Properties of Definition

1944 Descriptions for Part Properties of `Definition`:

1945 • `CellDefinition`

1946   semantic definition of a `Cell`.

1947   `CellDefinitions` groups one or more `CellDefinition` entities. See *Sec-*
1948   *tion 6.3.1 - CellDefinition*.

1949 • `Description`

1950   descriptive content.

1951   See *Section 4.1.2 - Description*.

1952 • `EntryDefinition`

1953   semantic definition of an `Entry`.

1954   `EntryDefinitions` groups one or more `EntryDefinition` entities. See
1955   *Section 6.3.2 - EntryDefinition*.

1956 6.3 Properties of Definition

1957 This section provides semantic information for the elements of the `Definition` for a
1958 `DataItem`.

1959 6.3.1 CellDefinition

1960 semantic definition of a `Cell`.

1961 6.3.1.1 Value Properties of CellDefinition

1962 *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

1963   Descriptions for Value Properties of `CellDefinition`:

1964   • units

1965   same as `DataItem` units. See *Section 6.1.1.1 - Value Properties of DataItem*.

1966   The value of units **MUST** be one of the `UnitEnum` enumeration.

1967   • key

1968   unique identification of the `Cell` in the `Definition`.

1969   The description applies to all `Cell` observations having this key.

1970   • type

1971   same as `DataItem` type. See *Section 6.5 - DataItem Types*.

1972   The value of type **MUST** be one of the `DataItemTypeEnum` enumeration.

1973   • subType

1974   same as `DataItem` subType. See *Section 6.1.1 - DataItem*.

1975   The value of subType **MUST** be one of the `DataItemSubTypeEnum` enumer-
1976   ation.

1977   • keyType

1978   `DataItem` type that defines the meaning of the key.

1979   The value of keyType **MUST** be one of the `DataItemTypeEnum` enumeration.

1980   **6.3.1.2   Part Properties of CellDefinition**

1981   *Table 83* lists the Part Properties of `CellDefinition`.

Part Property name	Multiplicity
Description	0..1

Table 83: Part Properties of CellDefinition

1982 Descriptions for Part Properties of `CellDefinition`:

1983 • `Description`

1984 descriptive content.

1985 See *Section 4.1.2 - Description.*

1986 6.3.2   EntryDefinition

1987 semantic definition of an `Entry`.

1988 6.3.2.1   Value Properties of EntryDefinition

1989 *Table 84* 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 84: Value Properties of EntryDefinition

1990 Descriptions for Value Properties of `EntryDefinition`:

1991 • `key`

1992 unique identification of the `Entry` in the `Definition`.

1993 The description applies to all `Entry` observations having this `key`.

1994 • `units`

1995 same as `DataItem units`. See *Section 6.1.1.1 - Value Properties of DataItem.*

1996 The value of `units` **MUST** be one of the `UnitEnum` enumeration.

1997 • `type`

1998   same as `DataItem type`. See *Section 6.5 - DataItem Types*.

1999   The value of `type` **MUST** be one of the `DataItemTypeEnum` enumeration.

2000 • `subType`

2001   same as `DataItem subType`. See *Section 6.1.1 - DataItem*.

2002   The value of `subType` **MUST** be one of the `DataItemSubTypeEnum` enumer-
2003   ation.

2004 • `keyType`

2005   `DataItem type` that defines the meaning of the `key`.

2006   The value of `keyType` **MUST** be one of the `DataItemTypeEnum` enumeration.

6.3.2.2 Part Properties of EntryDefinition

2008 *Table 85* lists the Part Properties of `EntryDefinition`.

Part Property name	Multiplicity
`Description`	0..1
`CellDefinition` (organized by `CellDefinitions`)	0..*

Table 85: Part Properties of EntryDefinition

2009 Descriptions for Part Properties of `EntryDefinition`:

2010 • `Description`

2011   descriptive content.

2012   See *Section 4.1.2 - Description*.

2013 • `CellDefinition`

2014   semantic definition of a `Cell`.

2015   `CellDefinitions` groups one or more `CellDefinition` entities if the rep-
2016   resentation, `DataItem` of `DataItem` is `TABLE`. See *Section 6.3.1 - CellDef-*
2017   *inition*.

6.4 Relationship Types for DataItem

2019 This section provides semantic information for the types of `AbstractDataItemRe-`
2020 `lationship` that can be defined for a `DataItem`.

2021 **6.4.1 AbstractDataItemRelationship**

2022 association between a `DataItem` and another entity.

2023 `AbstractDataItemRelationship` is an abstract entity and hence will be realized
2024 by specific `AbstractDataItemRelationship` types in an `MTConnectDevices`
2025 entity. See *Section 6.4 - Relationship Types for DataItem*.

2026 **6.4.1.1 Value Properties of AbstractDataItemRelationship**

2027 *Table 86* lists the Value Properties of `AbstractDataItemRelationship`.

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

Table 86: Value Properties of AbstractDataItemRelationship

2028 Descriptions for Value Properties of `AbstractDataItemRelationship`:

2029 • idRef

2030 reference to the related entity's `id`.

2031 • name

2032 descriptive name associated with this `AbstractDataItemRelationship`.

2033 **6.4.2 SpecificationRelationship**

2034 `AbstractDataItemRelationship` that provides a semantic reference to another
2035 `Specification` described by the `type` and `idRef` property.

2036 **6.4.2.1 Value Properties of SpecificationRelationship**

2037 *Table 87* lists the Value Properties of `SpecificationRelationship`.

Value Property name	Value Property type	Multiplicity
type	SpecificationRelationshipTypeEnum	1

Table 87: Value Properties of SpecificationRelationship

2038 Descriptions for Value Properties of `SpecificationRelationship`:

2039 • `type`

2040 specifies how the `Specification` is related.

2041 `SpecificationRelationshipTypeEnum` Enumeration:

2042 – `LIMIT`

2043 referenced `Specification` provides process limits.

6.4.3 DataItemRelationship

2044

2045 `AbstractDataItemRelationship` that provides a semantic reference to another
2046 `DataItem` described by the `type` property.

6.4.3.1 Value Properties of DataItemRelationship

2047

2048 *Table 88* lists the Value Properties of `DataItemRelationship`.

Value Property name	Value Property type	Multiplicity
type	DataItemRelationshipTypeEnum	1

Table 88: Value Properties of DataItemRelationship

2049 Descriptions for Value Properties of `DataItemRelationship`:

2050 • `type`

2051 specifies how the `DataItem` is related.

2052 `DataItemRelationshipTypeEnum` Enumeration:

2053 – `ATTACHMENT`

2054 reference to a `DataItem` that associates the values with an external entity.

2055   – COORDINATE_SYSTEM

2056   referenced `DataItem` provides the `id` of the effective Coordinate System.

2057   – LIMIT

2058   referenced `DataItem` provides process limits.

2059   – OBSERVATION

2060   referenced `DataItem` provides the observed values.

2061   6.5   DataItem Types

2062 This section provides semantic information for the types of a `DataItem`.

2063 In the MTConnect Standard, `DataItem` elements are defined and organized based upon
2064 the `category` and `type` attributes. The `category` attribute provides a high level
2065 grouping for `DataItem` elements based on the kind of information that is reported by
2066 the data item.

2067 These categories are:

2068   • `SAMPLE`: A `SAMPLE` reports a continuously variable or analog data value.

2069   • `EVENT`: An `EVENT` reports information representing a functional state, with two or
2070   more discrete values, associated with a component or it contains a message. The
2071   data provided may be a numeric value or text.

2072   • `CONDITION`: A `CONDITION` reports information about the health of a piece of
2073   equipment and its ability to function.

2074 The `type` attribute specifies the specific kind of data that is reported. For some types of
2075 data items, a `subType` attribute may also be used to differentiate between multiple data
2076 items of the same `type` where the information reported by the data item has a different,
2077 but related, meaning.

2078 Many types of data items provide two forms of data: a value (reported as either a `SAMPLE`
2079 or `EVENT`) and a health status (reported as a `CONDITION`). These `DataItem` types **MAY**
2080 be defined in more than one `category` based on the data that they report.

2081 6.5.1 Condition

2082 abstract `DataItem` that is about an entity's status regarding its ability to operate or it
2083 provides an indication whether the data reported for the entity is within an expected range.

2084 `Condition` is reported differently than `Sample` or `Event`. `Condition` **MUST** be
2085 reported as `Normal`, `Warning`, or `Fault`.

2086 All `Samples` **MAY** have associated `Condition` states. `Condition` states indicate
2087 whether the value for the data is within an expected range and **MUST** be reported as
2088 `Normal`, or the value is unexpected or out of tolerance for the data and a `Warning` or
2089 `Fault` **MUST** be provided.

2090 6.5.1.1 Value Properties of Condition

2091 *Table 89* lists the Value Properties of `Condition`.

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

Table 89: Value Properties of Condition

2092 Descriptions for Value Properties of `Condition`:

2093 • `type`
2094 `<<extensible>>` `ConditionEnum` Enumeration:

2095 – `ACTUATOR`
2096 indication of a fault associated with an actuator.
2097 – `COMMUNICATIONS`
2098 indication that the piece of equipment has experienced a communications fail-
2099 ure.
2100 – `DATA_RANGE`
2101 indication that the value of the data associated with a measured value or a
2102 calculation is outside of an expected range.
2103 – `LOGIC_PROGRAM`
2104 indication that an error occurred in the logic program or programmable logic
2105 controller (PLC) associated with a piece of equipment.

2106 – MOTION_PROGRAM
2107 indication that an error occurred in the motion program associated with a piece
2108 of equipment.

2109 – SYSTEM
2110 general purpose indication associated with an electronic component of a piece
2111 of equipment or a controller that represents a fault that is not associated with
2112 the operator, program, or hardware.

2113 6.5.2 Event

2114 abstract `DataItem` that is a discrete piece of information from a piece of equipment. It
2115 does not have intermediate values that vary over time.

2116 An `Event` is information that, when provided at any specific point in time, represents the
2117 current state of the piece of equipment.

2118 6.5.2.1 Value Properties of Event

2119 *Table 90* lists the Value Properties of `Event`.

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

Table 90: Value Properties of Event

2120 Descriptions for Value Properties of `Event`:

2121 • `type`
2122 `<<extensible>>` `EventEnum` Enumeration:

2123 – ACTIVATION_COUNT
2124 accumulation of the number of times a function has attempted to, or is planned
2125 to attempt to, activate or be performed.
2126 – ACTIVE_AXES
2127 set of axes currently associated with a `Path` or `Controller`.
2128 – ACTUATOR_STATE
2129 operational state of an apparatus for moving or controlling a mechanism or
2130 system.

2131 — ADAPTER_SOFTWARE_VERSION

2132 originator's software version of the *adapter*.

2133 — ADAPTER_URI

2134 Uniform Resource Identifier (URI) of the *adapter*.

2135 — <<deprecated>> ALARM

2136 **DEPRECATED:** Replaced with CONDITION category data items in Version
2137 1.1.0.

2138 — ALARM_LIMIT

2139 set of limits used to trigger warning or alarm indicators.

2140 — APPLICATION

2141 application on a Component.

2142 — ASSET_CHANGED

2143 assetId of the *Asset* that has been added or changed.

2144 — ASSET_COUNT

2145 *data set* of the number of *Assets* of a given type for a *Device*.

2146 — ASSET_REMOVED

2147 assetId of the *Asset* that has been removed.

2148 — AVAILABILITY

2149 *agent*'s ability to communicate with the data source.

2150 — AXIS_COUPLING

2151 describes the way the axes will be associated to each other.

2152 This is used in conjunction with COUPLED_AXES to indicate the way they are
2153 interacting.

2154 — AXIS_FEEDRATE_OVERRIDE

2155 value of a signal or calculation issued to adjust the feedrate of an individual
2156 linear type axis.

2157 — AXIS_INTERLOCK

2158 state of the axis lockout function when power has been removed and the axis
2159 is allowed to move freely.

2160 — AXIS_STATE

2161 state of a Linear or Rotary component representing an axis.

2162 — BATTERY_STATE

2163 present status of the battery.

2164 — BLOCK

2165 line of code or command being executed by a Controller entity.

2166 — BLOCK_COUNT

2167 total count of the number of blocks of program code that have been executed
2168 since execution started.

2169 — CHARACTERISTIC_PERSISTENT_ID

2170 UUID of the *characteristic*.

2171 — CHARACTERISTIC_STATUS

2172 pass/fail result of the measurement.

2173 — CHUCK_INTERLOCK

2174 state of an interlock function or control logic state intended to prevent the as-
2175 sociated `Chuck` component from being operated.

2176 — CHUCK_STATE

2177 operating state of a mechanism that holds a part or stock material during a
2178 manufacturing process.

2179 It may also represent a mechanism that holds any other mechanism in place
2180 within a piece of equipment.

2181 — CLOCK_TIME

2182 time provided by a timing device at a specific point in time.

2183 — <<deprecated>> CODE

2184 programmatic code being executed.

2185 **DEPRECATED** in *Version 1.1*.

2186 — COMPONENT_DATA

2187 `Event` that represents a `Component` where the `EntryDefinition` identi-
2188 fies the `Component` and the `CellDefinitions` define the `Component`'s
2189 observed `DataItems`.

2190 — COMPOSITION_STATE

2191 operating state of a mechanism represented by a `Composition` entity.

2192 — CONNECTION_STATUS

2193 status of the connection between an *adapter* and an *agent*.

2194 — CONTROLLER_MODE

2195 current mode of the `Controller` component.

2196 — CONTROLLER_MODE_OVERRIDE

2197 setting or operator selection that changes the behavior of a piece of equipment.

2198 — CONTROL_LIMIT

2199 set of limits used to indicate whether a process variable is stable and in control.

2200 — COUPLED_AXES

2201 set of associated axes.

2202   − CYCLE_COUNT

2203   accumulation of the number of times a cyclic function has attempted to, or is

2204   planned to attempt to execute.

2205   − DATE_CODE

2206   time and date code associated with a material or other physical item.

2207   − DEACTIVATION_COUNT

2208   accumulation of the number of times a function has attempted to, or is planned

2209   to attempt to, deactivate or cease.

2210   − DEVICE_ADDED

2211   UUID of new device added to an *MTConnect Agent*.

2212   − DEVICE_CHANGED

2213   UUID of the device whose *metadata* has changed.

2214   − DEVICE_REMOVED

2215   UUID of a device removed from an *MTConnect Agent*.

2216   − DEVICE_UUID

2217   identifier of another piece of equipment that is temporarily associated with a

2218   component of this piece of equipment to perform a particular function.

2219   − DIRECTION

2220   direction of motion.

2221   − DOOR_STATE

2222   operational state of a `Door` component or composition element.

2223   − EMERGENCY_STOP

2224   state of the emergency stop signal for a piece of equipment, controller path, or

2225   any other component or subsystem of a piece of equipment.

2226   − END_OF_BAR

2227   indication of whether the end of a piece of bar stock being feed by a bar feeder

2228   has been reached.

2229   − EQUIPMENT_MODE

2230   indication that a piece of equipment, or a sub-part of a piece of equipment, is

2231   performing specific types of activities.

2232   − EXECUTION

2233   execution status of the `Component`.

2234   − FEATURE_MEASUREMENT

2235   assessing elements of a *feature*.

2236   − FEATURE_PERSISTENT_ID

2237   UUID of a *feature*. *Ref ISO 10303 AP 242/239*.

2238 – FIRMWARE
2239   embedded software of a `Component` .

2240 – FIXTURE_ID
2241   identifier for a fixture.

2242 – FUNCTIONAL_MODE
2243   current intended production status of the `Component`.

2244 – HARDNESS
2245   hardness of a material.

2246 – HARDWARE
2247   hardware of a `Component`.

2248 – HOST_NAME
2249   name of the host computer supplying data.

2250 – LEAK_DETECT
2251   indication designating whether a leak has been detected.

2252 – LIBRARY
2253   software library on a `Component`

2254 – <<deprecated>> LINE
2255   current line of code being executed.
2256   **DEPRECATED** in *Version 1.4.0*.

2257 – LINE_LABEL
2258   identifier for a `Block` of code in a `Program`.

2259 – LINE_NUMBER
2260   position of a block of program code within a control program.

2261 – LOAD_COUNT
2262   accumulation of the number of times an operation has attempted to, or is
2263   planned to attempt to, load materials, parts, or other items.

2264 – LOCK_STATE
2265   state or operating mode of a `Lock`.

2266 – MAINTENANCE_LIST
2267   actions or activities to be performed in support of a piece of equipment.

2268 – MATERIAL
2269   identifier of a material used or consumed in the manufacturing process.

2270 – MATERIAL_LAYER
2271   identifies the layers of material applied to a part or product as part of an additive
2272   manufacturing process.

2273   – MEASUREMENT_TYPE

2274   class of measurement being performed. *Ref QIF 3:2018 Section 6.3*

2275   – MEASUREMENT_UNITS

2276   engineering units of the measurement.

2277   – MEASUREMENT_VALUE

2278   measurement based on the measurement type.

2279   – MESSAGE

2280   information to be transferred from a piece of equipment to a client software
2281   application.

2282   – MTCONNECT_VERSION

2283   reference version of the MTConnect Standard supported by the *adapter*.

2284   – NETWORK

2285   network details of a `Component`.

2286   – NETWORK_PORT

2287   number of the TCP/IP or UDP/IP port for the connection endpoint.

2288   – OPERATING_MODE

2289   state of `Component` or `Composition` that describes the automatic or man-
2290   ual operation of the entity.

2291   – OPERATING_SYSTEM

2292   Operating System (OS) of a `Component`.

2293   – OPERATOR_ID

2294   identifier of the person currently responsible for operating the piece of equip-
2295   ment.

2296   – PALLET_ID

2297   identifier for a pallet.

2298   – PART_COUNT

2299   aggregate count of parts.

2300   – PART_COUNT_TYPE

2301   interpretation of PART_COUNT.

2302   – PART_DETECT

2303   indication designating whether a part or work piece has been detected or is
2304   present.

2305   – PART_GROUP_ID

2306   identifier given to a collection of individual parts.

2307 — PART_ID

2308   identifier of a part in a manufacturing operation.

2309 — PART_KIND_ID

2310   identifier given to link the individual occurrence to a class of parts, typically

2311   distinguished by a particular part design.

2312 — <<deprecated>> PART_NUMBER

2313   identifier of a part or product moving through the manufacturing process.

2314   **DEPRECATED** in *Version 1.7*. PART_NUMBER is now a subType of PART_KIND_ID.

2315 — PART_PROCESSING_STATE

2316   particular condition of the part occurrence at a specific time.

2317 — PART_STATUS

2318   state or condition of a part.

2319 — PART_UNIQUE_ID

2320   identifier given to a distinguishable, individual part.

2321 — PATH_FEEDRATE_OVERRIDE

2322   value of a signal or calculation issued to adjust the feedrate for the axes asso-

2323   ciated with a Path component that may represent a single axis or the coordi-

2324   nated movement of multiple axes.

2325 — PATH_MODE

2326   describes the operational relationship between a Path entity and another Path

2327   entity for pieces of equipment comprised of multiple logical groupings of con-

2328   trolled axes or other logical operations.

2329 — POWER_STATE

2330   indication of the status of the source of energy for an entity to allow it to per-

2331   form its intended function or the state of an enabling signal providing permis-

2332   sion for the entity to perform its functions.

2333 — <<deprecated>> POWER_STATUS

2334   status of the Component.

2335   **DEPRECATED** in *Version 1.1.0*.

2336 — PROCESS_AGGREGATE_ID

2337   identifier given to link the individual occurrence to a group of related occur-

2338   rences, such as a process step in a process plan.

2339 — PROCESS_KIND_ID

2340   identifier given to link the individual occurrence to a class of processes or

2341   process definition.

2342 — PROCESS_OCCURRENCE_ID

2343   identifier of a process being executed by the device.

2344   – PROCESS_STATE
2345   particular condition of the process occurrence at a specific time.

2346   – PROCESS_TIME
2347   time and date associated with an activity or event.

2348   – PROGRAM
2349   name of the logic or motion program being executed by the `Controller`
2350   component.

2351   – PROGRAM_COMMENT
2352   comment or non-executable statement in the control program.

2353   – PROGRAM_EDIT
2354   indication of the status of the `Controller` components program editing
2355   mode.
2356   A program may be edited while another is executed.

2357   – PROGRAM_EDIT_NAME
2358   name of the program being edited.
2359   This is used in conjunction with `ProgramEdit` when in `ACTIVE` state.

2360   – PROGRAM_HEADER
2361   non-executable header section of the control program.

2362   – PROGRAM_LOCATION
2363   URI for the source file associated with `Program`.

2364   – PROGRAM_LOCATION_TYPE
2365   defines whether the logic or motion program defined by `Program` is being
2366   executed from the local memory of the controller or from an outside source.

2367   – PROGRAM_NEST_LEVEL
2368   indication of the nesting level within a control program that is associated with
2369   the code or instructions that is currently being executed.

2370   – ROTARY_MODE
2371   current operating mode for a `Rotary` type axis.

2372   – ROTARY_VELOCITY_OVERRIDE
2373   percentage change to the velocity of the programmed velocity for a `Rotary`
2374   axis.

2375   – ROTATION
2376   three space angular displacement of an object or coordinate system relative to
2377   a *cartesian coordinate system*.

2378   – SENSOR_ATTACHMENT
2379   *attachment* between a sensor and an entity.

2380   − SENSOR_STATE

2381   detection result of a sensor.

2382   − SERIAL_NUMBER

2383   serial number associated with a `Component`, `Asset`, or `Device`.

2384   − SPECIFICATION_LIMIT

2385   set of limits defining a range of values designating acceptable performance for

2386   a variable.

2387   − SPINDLE_INTERLOCK

2388   indication of the status of the spindle for a piece of equipment when power has

2389   been removed and it is free to rotate.

2390   − TOOL_ASSET_ID

2391   identifier of an individual tool asset.

2392   − TOOL_GROUP

2393   identifier for the tool group associated with a specific tool. Commonly used to

2394   designate spare tools.

2395   − <<deprecated>> TOOL_ID

2396   identifier of the tool currently in use for a given `Path`.

2397   **DEPRECATED** in *Version 1.2.0*. See `TOOL_ASSET_ID`.

2398   − TOOL_NUMBER

2399   identifier assigned by the `Controller` component to a cutting tool when in

2400   use by a piece of equipment.

2401   − TOOL_OFFSET

2402   reference to the tool offset variables applied to the active cutting tool.

2403   − TOOL_OFFSETS

2404   properties of each addressable tool offset.

2405   − TRANSFER_COUNT

2406   accumulation of the number of times an operation has attempted to, or is

2407   planned to attempt to, transfer materials, parts, or other items from one lo-

2408   cation to another.

2409   − TRANSLATION

2410   three space linear displacement of an object or coordinate system relative to a

2411   *cartesian coordinate system*.

2412   − UNCERTAINTY

2413   *uncertainty* specified by `UncertaintyType`.

2414   − UNCERTAINTY_TYPE

2415   method used to compute *standard uncertainty*.

2416   – `UNLOAD_COUNT`

2417   accumulation of the number of times an operation has attempted to, or is
2418   planned to attempt to, unload materials, parts, or other items.

2419   – `USER`

2420   identifier of the person currently responsible for operating the piece of equip-
2421   ment.

2422   – `VALVE_STATE`

2423   state of a valve is one of open, closed, or transitioning between the states.

2424   – `VARIABLE`

2425   data whose meaning may change over time due to changes in the operation of
2426   a piece of equipment or the process being executed on that piece of equipment.

2427   – `WAIT_STATE`

2428   indication of the reason that `Execution` is reporting a value of `WAIT`.

2429   – `WIRE`

2430   identifier for the type of wire used as the cutting mechanism in Electrical Dis-
2431   charge Machining or similar processes.

2432   – `WORKHOLDING_ID`

2433   identifier for the current workholding or part clamp in use by a piece of equip-
2434   ment.

2435   – `WORK_OFFSET`

2436   reference to offset variables for a work piece or part.

2437   – `WORK_OFFSETS`

2438   properties of each addressable work offset.

2439   6.5.3   Sample

2440   abstract `DataItem` that is continuously changing or analog data value.

2441   This data can be measured at any point-in-time and will always produce a result.

2442   The `units` for `Sample` **MUST** always be specified.

2443   6.5.3.1   Value Properties of Sample

2444   *Table 91* lists the Value Properties of `Sample`.

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

Table 91: Value Properties of Sample

2445 Descriptions for Value Properties of `Sample`:

2446 • `type`

2447 `<<extensible>> SampleEnum` Enumeration:

2448 **–** `ACCELERATION`
2449 positive rate of change of velocity.

2450 **–** `ACCUMULATED_TIME`
2451 accumulated time for an activity or event.

2452 **–** `<<deprecated>> AMPERAGE`
2453 strength of electrical current.
2454 **DEPRECATED** in *Version 1.6*. Replaced by `AMPERAGE_AC` and `AMPER-`
2455 `AGE_DC`.

2456 **–** `AMPERAGE_AC`
2457 electrical current that reverses direction at regular short intervals.

2458 **–** `AMPERAGE_DC`
2459 electric current flowing in one direction only.

2460 **–** `ANGLE`
2461 angular position.

2462 **–** `ANGULAR_ACCELERATION`
2463 positive rate of change of angular velocity.

2464 **–** `ANGULAR_DECELERATION`
2465 negative rate of change of angular velocity.

2466 **–** `ANGULAR_VELOCITY`
2467 rate of change of angular position.

2468 **–** `ASSET_UPDATE_RATE`
2469 average rate of change of values for assets in the MTConnect streams.
2470 The average is computed over a rolling window defined by the implementation.

2471 **–** `AXIS_FEEDRATE`
2472 feedrate of a linear axis.

2473   – BATTERY_CAPACITY

2474   maximum rated charge a battery is capable of maintaining based on the battery
2475   discharging at a specified current over a specified time period.

2476   – BATTERY_CHARGE

2477   value of the battery's present capacity expressed as a percentage of the battery's
2478   maximum rated capacity.

2479   – CAPACITY_FLUID

2480   fluid capacity of an object or container.

2481   – CAPACITY_SPATIAL

2482   geometric capacity of an object or container.

2483   – CHARGE_RATE

2484   value of the current being supplied to the Component for the purpose of
2485   charging.

2486   – CONCENTRATION

2487   percentage of one component within a mixture of components.

2488   – CONDUCTIVITY

2489   ability of a material to conduct electricity.

2490   – CUTTING_SPEED

2491   speed difference (relative velocity) between the cutting mechanism and the
2492   surface of the workpiece it is operating on.

2493   – DECELERATION

2494   negative rate of change of velocity.

2495   – DENSITY

2496   volumetric mass of a material per unit volume of that material.

2497   – DEPOSITION_ACCELERATION_VOLUMETRIC

2498   rate of change in spatial volume of material deposited in an additive manufac-
2499   turing process.

2500   – DEPOSITION_DENSITY

2501   density of the material deposited in an additive manufacturing process per unit
2502   of volume.

2503   – DEPOSITION_MASS

2504   mass of the material deposited in an additive manufacturing process.

2505   – DEPOSITION_RATE_VOLUMETRIC

2506   rate at which a spatial volume of material is deposited in an additive manufac-
2507   turing process.

2508   – `DEPOSITION_VOLUME`
2509   spatial volume of material to be deposited in an additive manufacturing pro-
2510   cess.

2511   – `DEW_POINT`
2512   temperature at which moisture begins to condense, corresponding to saturation
2513   for a given absolute humidity.

2514   – `DIAMETER`
2515   dimension of a diameter.

2516   – `DISCHARGE_RATE`
2517   value of current being drawn from the `Component`.

2518   – `DISPLACEMENT`
2519   change in position of an object.

2520   – `DISPLACEMENT_ANGULAR`
2521   absolute value of the change in angular position around a vector

2522   – `DISPLACEMENT_LINEAR`
2523   absolute value of the change in position along a vector.

2524   – `ELECTRICAL_ENERGY`
2525   `Wattage` used or generated by a component over an interval of time.

2526   – `EQUIPMENT_TIMER`
2527   amount of time a piece of equipment or a sub-part of a piece of equipment has
2528   performed specific activities.

2529   – `FILL_LEVEL`
2530   amount of a substance remaining compared to the planned maximum amount
2531   of that substance.

2532   – `FLOW`
2533   rate of flow of a fluid.

2534   – `FOLLOWING_ERROR`
2535   difference between actual and commanded position at any specific point in
2536   time during a motion.

2537   – `FOLLOWING_ERROR_ANGULAR`
2538   angular difference between the commanded encoder/resolver position and the
2539   actual encoder/resolver position at any specified point in time during a motion.

2540   – `FOLLOWING_ERROR_LINEAR`
2541   difference between the commanded encoder/resolver position and the actual
2542   encoder/resolver position at any specified point in time during a motion.

2543 – FREQUENCY

2544 number of occurrences of a repeating event per unit time.

2545 – <<deprecated>> GLOBAL_POSITION

2546 position in three-dimensional space.

2547 **DEPRECATED** in Version 1.1.

2548 – GRAVITATIONAL_ACCELERATION

2549 acceleration relative to Earth's gravity of 9.80665 METER/SECOND^2.

2550 – GRAVITATIONAL_FORCE

2551 force relative to earth's gravity.

2552 – HUMIDITY_ABSOLUTE

2553 amount of water vapor expressed in grams per cubic meter.

2554 – HUMIDITY_RELATIVE

2555 amount of water vapor present expressed as a percent to reach saturation at the
2556 same temperature.

2557 – HUMIDITY_SPECIFIC

2558 ratio of the water vapor present over the total weight of the water vapor and air
2559 present expressed as a percent.

2560 – LENGTH

2561 length of an object.

2562 – <<deprecated>> LEVEL

2563 level of a resource.

2564 **DEPRECATED** in *Version 1.2*. See FILL_LEVEL.

2565 – LINEAR_FORCE

2566 *force* applied to a mass in one direction only.

2567 – LOAD

2568 actual versus the standard rating of a piece of equipment.

2569 – MASS

2570 mass of an object(s) or an amount of material.

2571 – OBSERVATION_UPDATE_RATE

2572 average rate of change of values for data items in the MTConnect streams. The
2573 average is computed over a rolling window defined by the implementation.

2574 – OPENNESS

2575 percentage open where 100% is fully open and 0% is fully closed.

2576 – ORIENTATION

2577 angular position of a plane or vector relative to a *cartesian coordinate system*

2578   − PATH_FEEDRATE

2579   feedrate for the axes, or a single axis, associated with a `Path` component.

2580   − PATH_FEEDRATE_PER_REVOLUTION

2581   feedrate for the axes, or a single axis.

2582   − PATH_POSITION

2583   position of a control point associated with a `Controller` or a `Path`.

2584   − PH

2585   acidity or alkalinity of a solution.

2586   − POSITION

2587   point along an axis in a *cartesian coordinate system*.

2588   − POSITION_CARTESIAN

2589   point in a *cartesian coordinate system*.

2590   − POWER_FACTOR

2591   ratio of real power flowing to a load to the apparent power in that AC circuit.

2592   − PRESSURE

2593   force per unit area measured relative to atmospheric pressure.

2594   Commonly referred to as gauge pressure.

2595   − PRESSURE_ABSOLUTE

2596   force per unit area measured relative to a vacuum.

2597   − PRESSURIZATION_RATE

2598   change of pressure per unit time.

2599   − PROCESS_TIMER

2600   amount of time a piece of equipment has performed different types of activities
2601   associated with the process being performed at that piece of equipment.

2602   − RESISTANCE

2603   degree to which a substance opposes the passage of an electric current.

2604   − ROTARY_VELOCITY

2605   rotational speed of a rotary axis.

2606   − SETTLING_ERROR

2607   difference between actual and commanded position at the end of a motion.

2608   − SETTLING_ERROR_ANGULAR

2609   angular difference between the commanded encoder/resolver position, and the
2610   actual encoder/resolver position when motion is complete.

2611   − SETTLING_ERROR_LINEAR

2612   difference between the commanded encoder/resolver position, and the actual
2613   encoder/resolver position when motion is complete.

2614   − SOUND_LEVEL

2615   sound level or sound pressure level relative to atmospheric pressure.

2616   − <<deprecated>> SPINDLE_SPEED

2617   rotational speed of the rotary axis.

2618   **DEPRECATED** in *Version 1.2*. Replaced by ROTARY_VELOCITY.

2619   − STRAIN

2620   amount of deformation per unit length of an object when a load is applied.

2621   − TEMPERATURE

2622   degree of hotness or coldness measured on a definite scale.

2623   − TENSION

2624   force that stretches or elongates an object.

2625   − TILT

2626   angular displacement.

2627   − TORQUE

2628   turning force exerted on an object or by an object.

2629   − VELOCITY

2630   rate of change of position of a Component.

2631   − VISCOSITY

2632   fluid's resistance to flow.

2633   − <<deprecated>> VOLTAGE

2634   electrical potential between two points.

2635   **DEPRECATED** in *Version 1.6*. Replaced by VOLTAGE_AC and VOLTAGE_DC.

2636   − VOLTAGE_AC

2637   electrical potential between two points in an electrical circuit in which the cur-
2638   rent periodically reverses direction.

2639   − VOLTAGE_DC

2640   electrical potential between two points in an electrical circuit in which the cur-
2641   rent is unidirectional.

2642   − VOLT_AMPERE

2643   apparent power in an electrical circuit, equal to the product of root-mean-
2644   square (RMS) voltage and RMS current (commonly referred to as VA).

2645   − VOLT_AMPERE_REACTIVE

2646   reactive power in an AC electrical circuit (commonly referred to as VAR).

2647   − VOLUME_FLUID

2648   fluid volume of an object or container.

2649   – VOLUME_SPATIAL

2650   geometric volume of an object or container.

2651   – WATTAGE

2652   power flowing through or dissipated by an electrical circuit or piece of equip-
2653   ment.

2654   – X_DIMENSION

2655   dimension of an entity relative to the X direction of the referenced coordinate
2656   system.

2657   – Y_DIMENSION

2658   dimension of an entity relative to the Y direction of the referenced coordinate
2659   system.

2660   – Z_DIMENSION

2661   dimension of an entity relative to the Z direction of the referenced coordinate
2662   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 92 lists the Value Properties of `Reference`.

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

**Table 92:** Value Properties of Reference
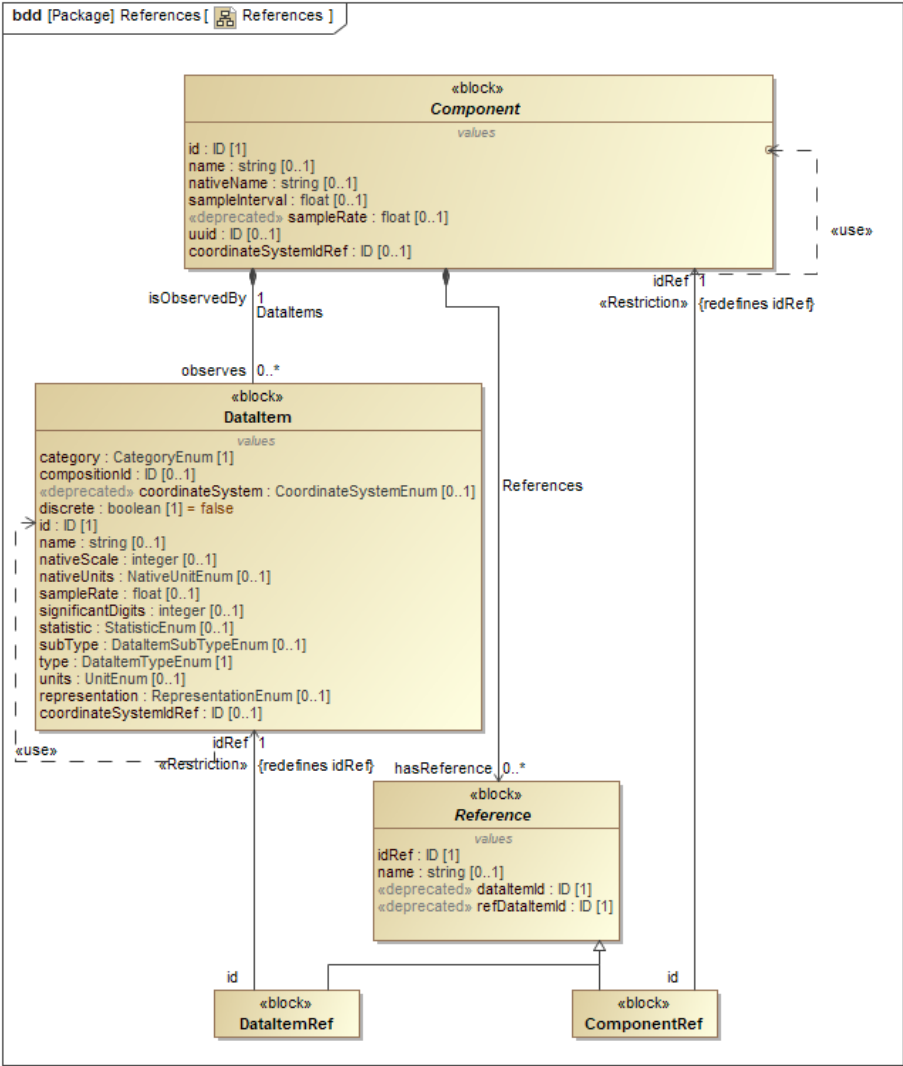
Figure 6: References

2681 Descriptions for Value Properties of `Reference`:

2682 • `idRef`

2683 pointer to the `id` of an entity that contains the information to be associated with this
2684 entity.

2685 • `name`

2686 name of an element or a piece of equipment.

2687   7.1.2   DataItemRef

2688 `Reference` that is a pointer to a `DataItem` associated with another entity defined for
2689 a piece of equipment.

2690 `DataItemRef` allows the data associated with a `DataItem` defined in another entity to
2691 be directly associated with this entity.

2692   7.1.3   ComponentRef

2693 `Reference` that is a pointer to all of the information associated with another entity de-
2694 fined for a piece of equipment.

2695 `ComponentRef` allows all of the information of (*lower level* `Component` entities that
2696 is associated with the other entity to be directly associated with this entity.

8 Configurations Model

2697

2698 This section provides semantic information for the `Configuration` entity that is used
2699 to model technical information about a `Component`.

8.1 Configurations

2700

2701 Figure 7 shows the abstract `Configuration` and its types.
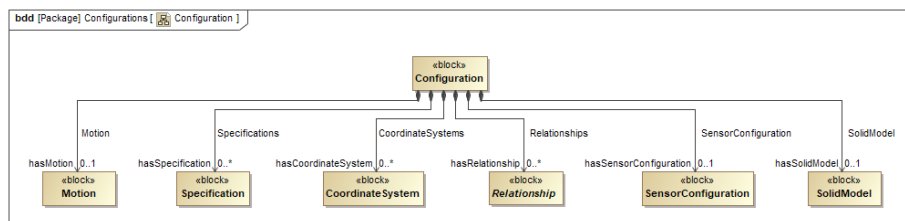
Figure 7: Configuration

2702   Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

8.1.1 Configuration

2703

2704 technical information about an entity describing its physical layout, functional character-
2705 istics, and relationships with other entities.

8.1.1.1 Part Properties of Configuration

2706

2707 *Table 93* 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..*

Table 93: Part Properties of Configuration

2708 Descriptions for Part Properties of Configuration:

2709 • SolidModel

2710 references to a file with the three-dimensional geometry of the Component or
2711 Composition.

2712 See *Section 8.7 - SolidModel*.

2713 • SensorConfiguration

2714 configuration for a Sensor.

2715 See SensorConfiguration.

2716 • Motion

2717 movement of the component relative to a coordinate system.

2718 See *Section 8.4 - Motion*.

2719 • ConfigurationRelationship

2720 association between two pieces of equipment that function independently but to-
2721 gether perform a manufacturing operation.

2722 Relationships groups one or more ConfigurationRelationship types.
2723 See *Section 8.5 - Relationships*.

2724 • CoordinateSystem

2725 reference system that associates a unique set of n parameters with each point in an
2726 n-dimensional space. *Ref ISO 10303-218:2004*

2727 CoordinateSystems groups one or more CoordinateSystem entities. See
2728 *Section 8.2 - CoordinateSystems*.

2729 • `Specification`

2730 design characteristics for a piece of equipment.

2731 `Specifications` groups one or more `Specification` entities. See *Sec-*
2732 *tion 8.8 - Specifications*.

2733 • `ImageFile`

2734 reference to a file containing an image of the `Component`.

2735 `ImageFiles` groups one or more `ImageFile` entities. See *Section 8.3 - Image-*
2736 *Files*.

2737 8.2   CoordinateSystems

2738 This section provides semantic information for the `CoordinateSystem` entity.

2739   Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

2740   Note: See *Example 4* for the XML representation of the same example.

2741 8.2.1   CoordinateSystem

2742 reference system that associates a unique set of n parameters with each point in an n-
2743 dimensional space. *Ref ISO 10303-218:2004*

2744 At most only one of `Origin` or `Transformation` **MUST** be defined for a `Coordi-`
2745 `nateSystem`.

2746 8.2.1.1   Value Properties of CoordinateSystem

2747 *Table 94* 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	ID	0..1
Description	string	0..1

Table 94: Value Properties of CoordinateSystem
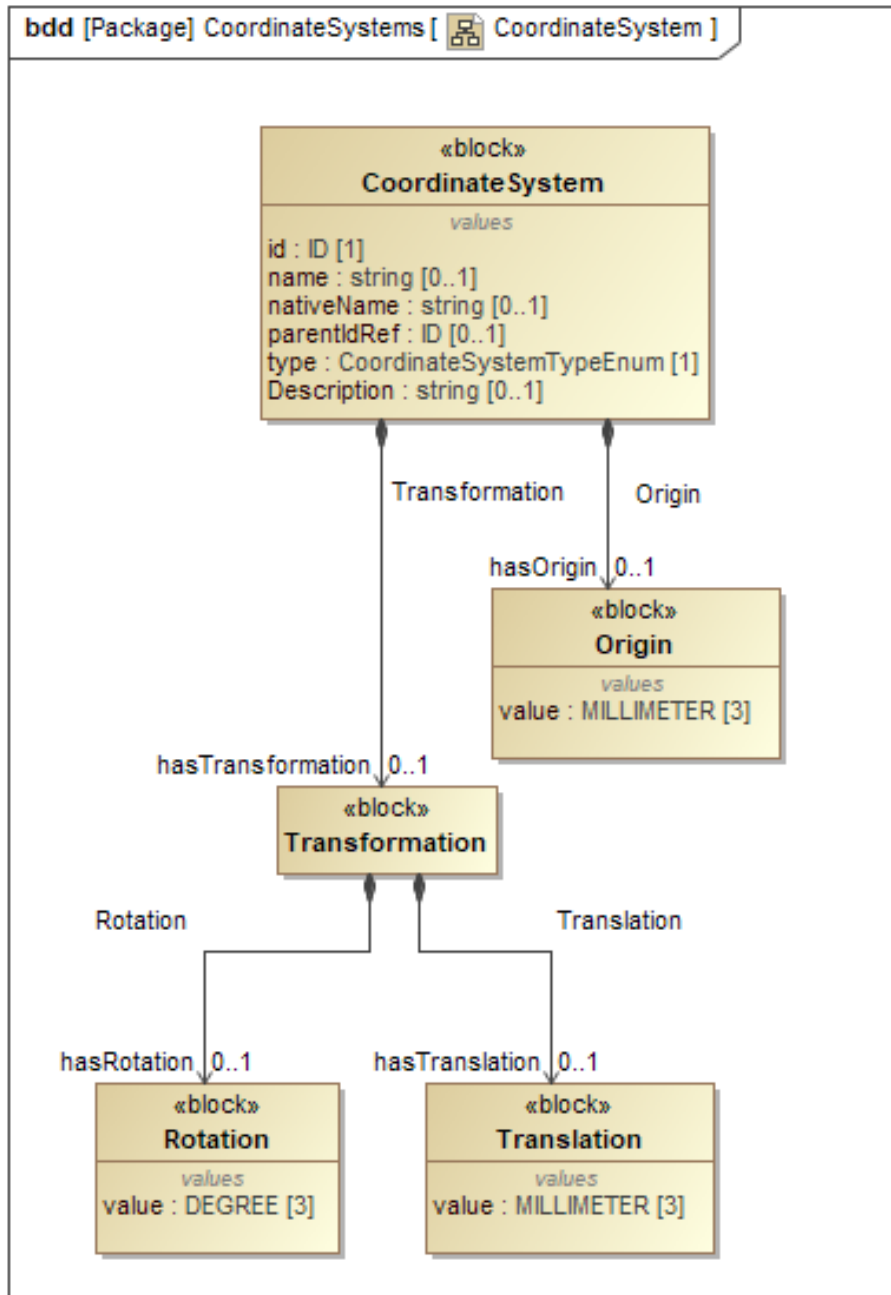
Figure 8: CoordinateSystem
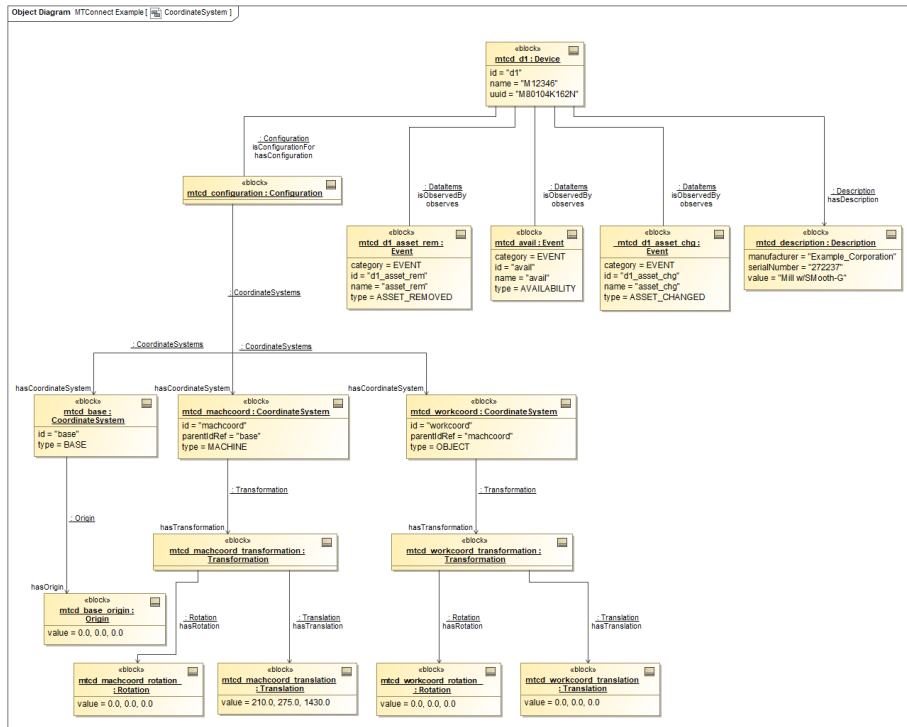
**Figure 9:** CoordinateSystem Example

2748 Descriptions for Value Properties of `CoordinateSystem`:

2749 • `id`

2750 unique identifier for the coordinate system.

2751 • `name`

2752 name of the coordinate system.

2753 • `nativeName`

2754 manufacturer's name or users name for the coordinate system.

2755 • `parentIdRef`

2756 pointer to the `id` attribute of the parent `CoordinateSystem`.

2757 • `type`

2758 type of coordinate system.

2759 `CoordinateSystemTypeEnum` Enumeration:

2760 – `BASE`

2761 coordinate system referenced to the base mounting surface. *Ref ISO 9787:2013*

2762 A base mounting surface is a connection surface between the arm and its sup-

2763 porting structure.*Ref ISO 9787:2013*

2764 For non-robotic devices, it is the connection surface between the device and its

2765 supporting structure.

2766 – `CAMERA`

2767 coordinate system referenced to the sensor which monitors the site of the task.

2768 *Ref ISO 9787:2013*

2769 – `MACHINE`

2770 coordinate system referenced to the home position and orientation of the pri-

2771 mary axes of a piece of equipment.

2772 – `MECHANICAL_INTERFACE`

2773 coordinate system referenced to the mechanical interface. *Ref ISO 9787:2013*

2774 – `MOBILE_PLATFORM`

2775 coordinate system referenced to one of the components of a mobile platform.

2776 *Ref ISO 8373:2012*

2777 – `OBJECT`

2778 coordinate system referenced to the object. *Ref ISO 9787:2013*

2779   – `TASK`

2780   coordinate system referenced to the site of the task. *Ref ISO 9787:2013*

2781   – `TOOL`

2782   coordinate system referenced to the tool or to the end effector attached to the
2783   mechanical interface. *Ref ISO 9787:2013*

2784   – `WORLD`

2785   stationary coordinate system referenced to earth, which is independent of the
2786   robot motion. *Ref ISO 9787:2013*

2787   For non-robotic devices, stationary coordinate system referenced to earth, which
2788   is independent of the motion of a piece of equipment.

2789   • `uuid`

2790   UUID for the coordinate system.

2791   • `Description`

2792   natural language description of the `CoordinateSystem`.

8.2.1.2   Part Properties of CoordinateSystem

2794   *Table 95* lists the Part Properties of `CoordinateSystem`.

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

Table 95: Part Properties of CoordinateSystem

2795   Descriptions for Part Properties of `CoordinateSystem`:

2796   • `Origin`

2797   coordinates of the origin position of a coordinate system.

2798   See *Section 8.2.2 - Origin*.

2799   • `Transformation`

2800   process of transforming to the origin position of the coordinate system from a parent
2801   coordinate system using `Translation` and `Rotation`.

2802   See *Section 8.2.3 - Transformation*.

2803 ## 8.2.2 Origin

2804 coordinates of the origin position of a coordinate system.

2805 The value of `Origin` **MUST** be reported in `MILLIMETER_3D`.

2806 ## 8.2.3 Transformation

2807 process of transforming to the origin position of the coordinate system from a parent co-
2808 ordinate system using `Translation` and `Rotation`.

2809 At a minimum, a `Translation` or a `Rotation` **MUST** be defined for a `Transfor-`
2810 `mation`.

2811 ### 8.2.3.1 Part Properties of Transformation

2812 *Table 96* lists the Part Properties of `Transformation`.

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

Table 96: Part Properties of Transformation

2813 Descriptions for Part Properties of `Transformation`:

2814   • `Translation`

2815   translations along X, Y, and Z axes are expressed as x,y, and z respectively within a
2816   3-dimensional vector.

2817   See *Section 8.2.5 - Translation*.

2818   • `Rotation`

2819   rotations about X, Y, and Z axes are expressed in A, B, and C respectively within a
2820   3-dimensional vector.

2821   See *Section 8.2.4 - Rotation*.

2822 8.2.4   Rotation

2823 rotations about X, Y, and Z axes are expressed in A, B, and C respectively within a 3-
2824 dimensional vector.

2825 The value of `Rotation` **MUST** be reported in `DEGREE_3D`.

2826 8.2.5   Translation

2827 translations along X, Y, and Z axes are expressed as x,y, and z respectively within a 3-
2828 dimensional vector.

2829 The value of `Translation` **MUST** be reported in `MILLIMETER_3D`.

2830 8.3   ImageFiles

2831 This section provides semantic information for the `ImageFile` entity.

2832 8.3.1   ImageFile

2833 reference to a file containing an image of the `Component`.

2834 8.3.1.1   Value Properties of ImageFile

2835 *Table 97* 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 97: Value Properties of ImageFile

2836 Descriptions for Value Properties of `ImageFile`:

2837　• `id`

2838　　unique identifier of the image file.

2839　• `href`

2840　　Uniform Resource Locator (URL) giving the location of the image file.

2841　• `mediaType`

2842　　mime type of the image file.

2843　• `name`

2844　　description of the image file.

2845　8.4　Motion

2846　This section provides semantic information for the `Motion` entity.
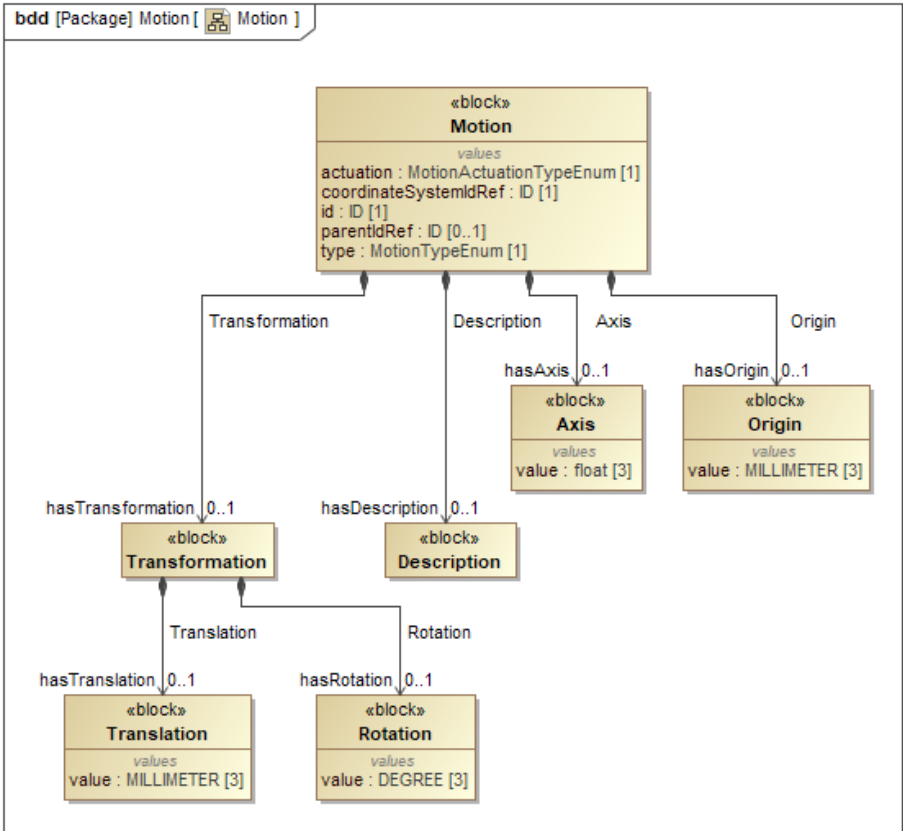

Figure 10: Motion

2847 Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.
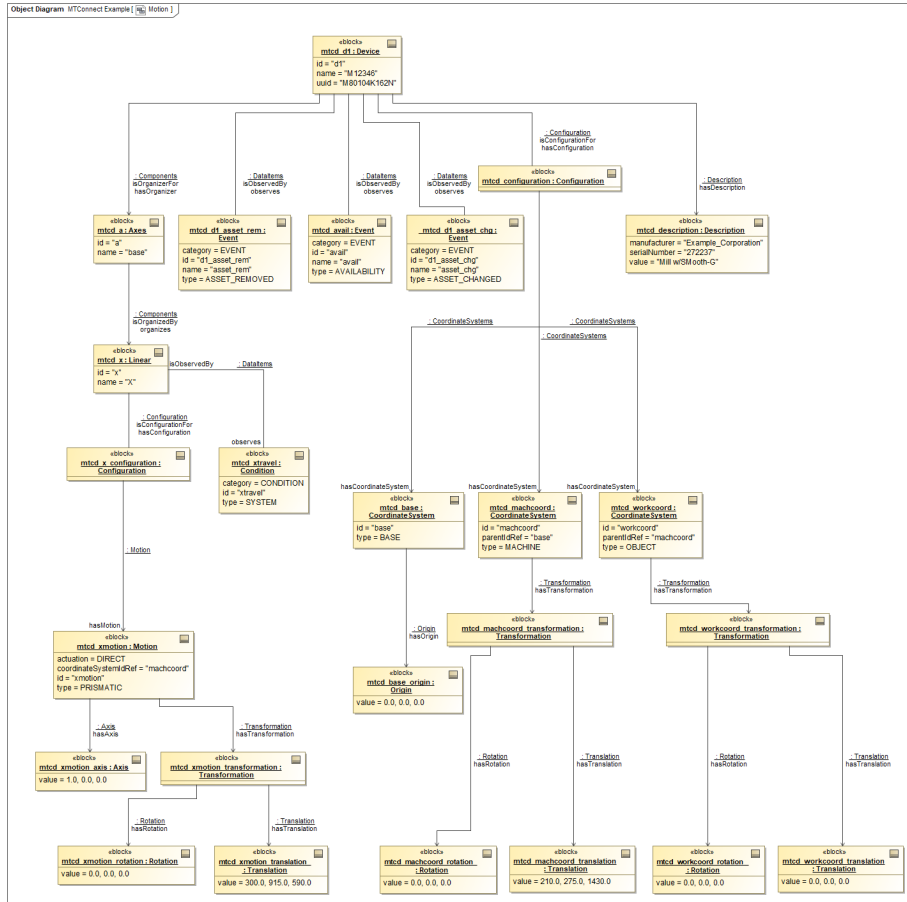
Figure 11: Motion Example

2848 Note: See *Example 5* for the XML representation of the same example.

8.4.1 Motion

2850 movement of the component relative to a coordinate system.

2851 `Motion` specifies the kinematic chain of the `component` entities.

2852 At most only one of `Origin` or `Transformation` **MUST** be defined for a `Motion`.

2853 **8.4.1.1   Value Properties of Motion**

2854 *Table 98* lists the Value Properties of `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 98: Value Properties of Motion

2855 Descriptions for Value Properties of `Motion`:

2856   • `actuation`

2857   describes if this component is actuated directly or indirectly as a result of other
2858   motion.

2859   `MotionActuationTypeEnum` Enumeration:

2860   – `DIRECT`
2861   movement is initiated by the component.
2862   – `NONE`
2863   no actuation of this axis.

2864   Note: Actuation of `NONE` can be either a derived `REVOLUTE` or
2865   `PRISMATIC` motion or static `FIXED` relationship.

2866   – `VIRTUAL`
2867   motion is computed and is used for expressing an imaginary movement.

2868   • `coordinateSystemIdRef`

2869   coordinate system within which the kinematic motion occurs.

2870   • `id`

2871   unique identifier for this element.

2872   • `parentIdRef`

2873   pointer to the `id` attribute of the parent `Motion`.

2874   The kinematic chain connects all components using the parent relations. All motion
2875   is connected to the motion of the parent. The first node in the chain will not have a
2876   parent.

2877   • `type`

2878   type of motion.

2879   `MotionTypeEnum` Enumeration:

2880   – `CONTINUOUS`
2881   revolves around an axis with a continuous range of motion.
2882   – `FIXED`
2883   axis does not move.
2884   – `PRISMATIC`
2885   sliding linear motion along an axis with a fixed range of motion.
2886   – `REVOLUTE`
2887   rotates around an axis with a fixed range of motion.

2888   ### 8.4.1.2   Part Properties of Motion

2889   *Table 99* lists the Part Properties of `Motion`.

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

Table 99: Part Properties of Motion

2890   Descriptions for Part Properties of `Motion`:

2891   • `Axis`

2892   axis along or around which the `Component` moves relative to a coordinate system.

2893   See *Section 4.3.5 - Axis*.

2894   • `Origin`

2895   coordinates of the origin position of a coordinate system.

2896   See *Section 8.2.2 - Origin*.

2897   • `Transformation`

2898   process of transforming to the origin position of the coordinate system from a parent
2899   coordinate system using `Translation` and `Rotation`.

2900   See *Section 8.2.3 - Transformation*.

2901 • `Description`

2902   descriptive content.

2903   See *Section 4.1.2 - Description*.

2904 8.4.2 Axis

2905 axis along or around which the `Component` moves relative to a coordinate system.

2906 The value of `Axis` **MUST** be a list of `float` of size 3.

2907 8.5 Relationships

2908 This section provides semantic information for the `ConfigurationRelationship`
2909 entity.
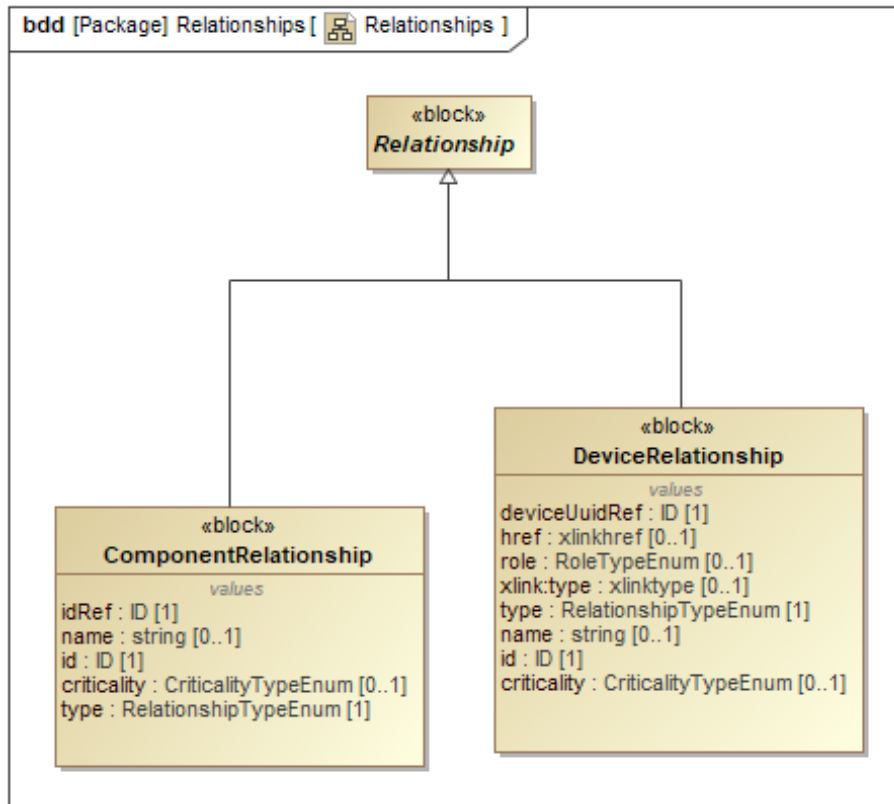
Figure 12: Relationships

2910   Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.
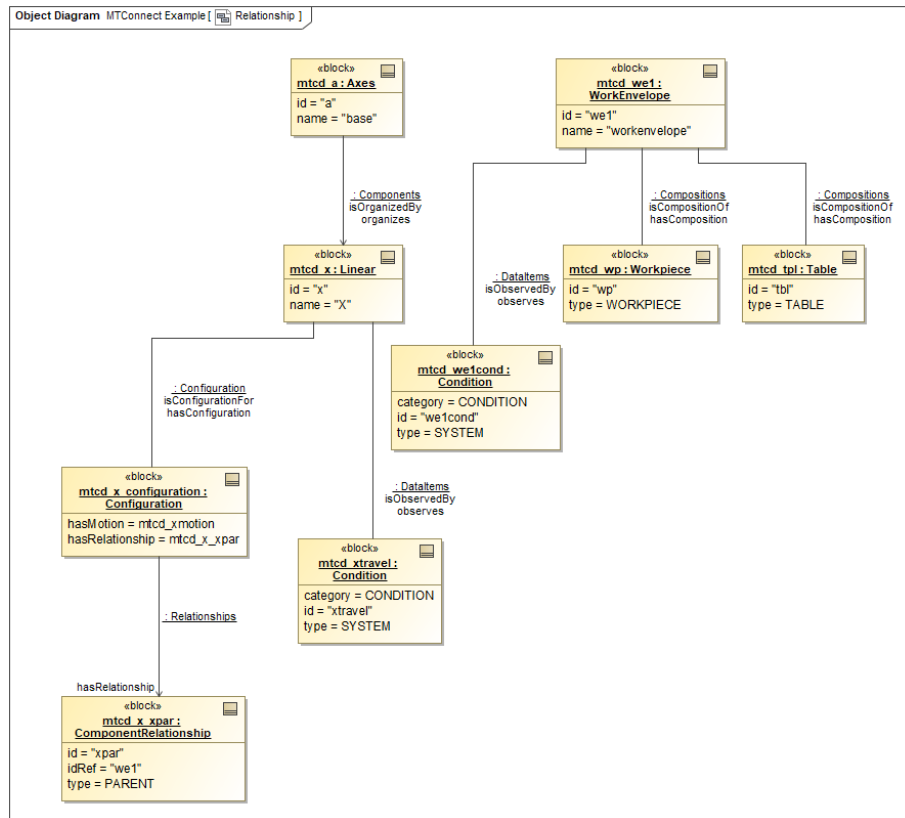
Figure 13: Relationship Example

2911   Note: See *Example 6* for the XML representation of the same example.

2912   8.5.1   ConfigurationRelationship

2913 association between two pieces of equipment that function independently but together
2914 perform a manufacturing operation.

2915 ConfigurationRelationship is an abstract entity and hence will be realized by
2916 specific ConfigurationRelationship types in an MTConnectDevices entity.
2917 See *Section 8.5.2 - ComponentRelationship* and *Section 8.5.3 - DeviceRelationship*.

2918   8.5.1.1   Value Properties of ConfigurationRelationship

2919 *Table 100* 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 100: Value Properties of ConfigurationRelationship

2920 Descriptions for Value Properties of `ConfigurationRelationship`:

2921 • `name`

2922 name associated with this `ConfigurationRelationship`.

2923 • `id`

2924 unique identifier for this `ConfigurationRelationship`.

2925 • `type`

2926 defines the authority that this piece of equipment has relative to the associated piece
2927 of equipment.

2928 `RelationshipTypeEnum` Enumeration:

2929 – `CHILD`
2930 functions as a child in the relationship with the associated element.

2931 – `PARENT`
2932 functions as a parent in the relationship with the associated element.

2933 – `PEER`
2934 functions as a peer which provides equal functionality and capabilities in the
2935 relationship with the associated element.

2936 • `criticality`

2937 defines whether the services or functions provided by the associated piece of equip-
2938 ment is required for the operation of this piece of equipment.

2939 `CriticalityTypeEnum` Enumeration:

2940 – `CRITICAL`
2941 services or functions provided by the associated element is required for the
2942 operation of this element.

2943 – `NONCRITICAL`
2944 services or functions provided by the associated element is not required for the
2945 operation of this element.

2946 8.5.2 ComponentRelationship

2947 `ConfigurationRelationship` that describes the association between two compo-
2948 nents within a piece of equipment that function independently but together perform a ca-
2949 pability or service within a piece of equipment.

2950 8.5.2.1 Value Properties of ComponentRelationship

2951 *Table 101* lists the Value Properties of `ComponentRelationship`.

Value Property name	Value Property type	Multiplicity
idRef	ID	1

Table 101: Value Properties of ComponentRelationship

2952 Descriptions for Value Properties of `ComponentRelationship`:

2953 • `idRef`

2954   reference to the associated `Component`.

2955 8.5.3 DeviceRelationship

2956 `ConfigurationRelationship` that describes the association between two pieces of
2957 equipment that function independently but together perform a manufacturing operation.

2958 8.5.3.1 Value Properties of DeviceRelationship

2959 *Table 102* lists the Value Properties of `DeviceRelationship`.

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

Table 102: Value Properties of DeviceRelationship

2960 Descriptions for Value Properties of `DeviceRelationship`:

2961 • deviceUuidRef

2962 reference to the uuid attribute of the Device element of the associated piece of
2963 equipment.

2964 • href

2965 URI identifying the *agent* that is publishing information for the associated piece of
2966 equipment.

2967 • role

2968 defines the services or capabilities that the referenced piece of equipment provides
2969 relative to this piece of equipment.

2970 RoleTypeEnum Enumeration:

2971 – AUXILIARY

2972 associated element performs the functions as an Auxiliary for this element.

2973 – SYSTEM

2974 associated element performs the functions of a System for this element.

2975 • xlink:type

2976 xlink:type**MUST** have a fixed value of locator as defined in W3C XLink
2977 1.1 *Ref https://www.w3.org/TR/xlink11/.*

2978 8.5.4 AssetRelationship

2979 ConfigurationRelationship that describes the association between a Compo-
2980 nent and an Asset.

2981 8.5.4.1 Value Properties of AssetRelationship

2982 *Table 103* lists the Value Properties of AssetRelationship.

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

Table 103: Value Properties of AssetRelationship

2983 Descriptions for Value Properties of AssetRelationship:

2984   • `assetIdRef`

2985   uuid of the related `Asset`.

2986   • `assetType`

2987   type of `Asset` being referenced.

2988   • `href`

2989   URI reference to the associated `Asset`.

2990   8.6   Sensor

2991 This section provides semantic information for the `SensorConfiguration` entity.

2992 sensor is a unique type of a piece of equipment. A sensor is typically comprised of two
2993 major components: a sensor unit that provides signal processing, conversion, and commu-
2994 nications and the *sensing elements* that provides a signal or measured value.

2995 The sensor unit is modeled as a *lower level* `Component` called `Sensor`. The *sensing*
2996 *element* may be modeled as a `Composition` element of a `Sensor` element and the
2997 measured value would be modeled as a `DataItem` (See *Section 6.1 - DataItems* for more
2998 information on `DataItem` elements). Each sensor unit may have multiple *sensing ele-*
2999 *ments*; each representing the data for a variety of measured values.

3000   Note: See *Example 9* for an XML example.

3001 When a sensor unit is modeled as a `Component` or as a separate piece of equipment, it
3002 may provide additional configuration information for the sensor elements and the sensor
3003 unit itself.

3004   Note: If a `Sensor` provides vibration measurement data for the spindle on a
3005   piece of equipment, it could be modeled as a `Sensor` for rotary axis named
3006   `C`. See *Example 10* for an XML example.

3007   Note: If a `Sensor` provides measurement data for multiple `Component` el-
3008   ements within a piece of equipment and is not associated with any particular
3009   `Component`, it **MAY** be modeled as an independent `Component` and the
3010   data associated with measurements are associated with their associated `Com-`
3011   `ponent` entities. See *Example 11* for an XML example.
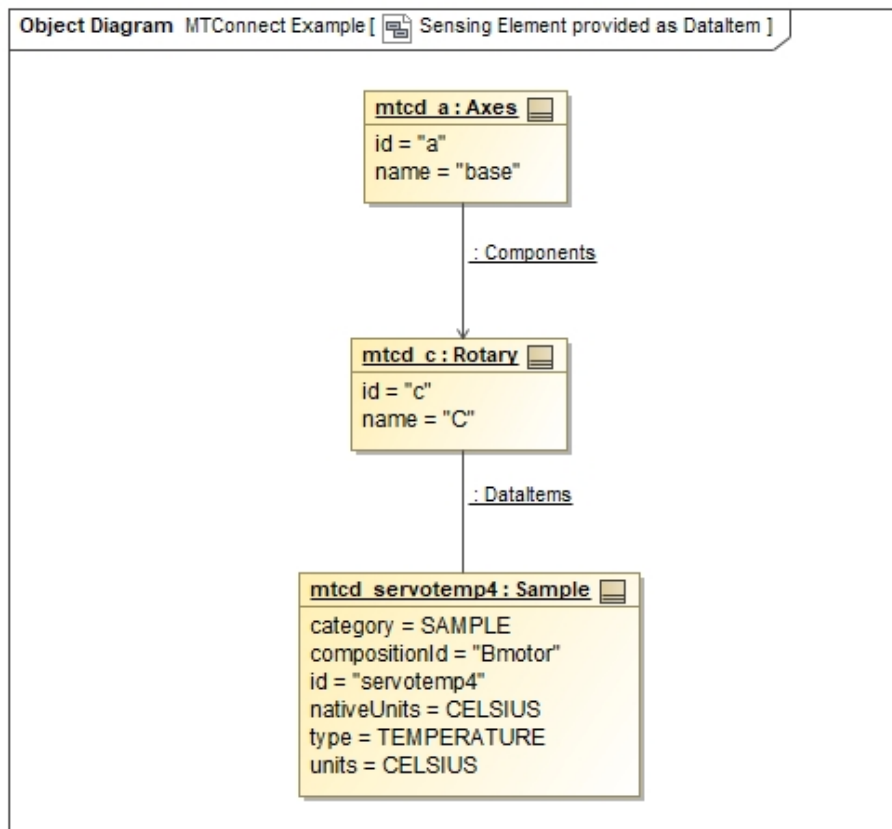
Figure 14: Sensing Element provided as a DataItem Example

3012 `Configuration` data provides information required for maintenance and support of the
3013 sensor.

3014 When `Sensor` represents the sensor unit for multiple *sensing element*(s), each sensing
3015 element is represented by a `Channel`. The sensor unit itself and each `Channel` repre-
3016 senting one *sensing element* **MAY** have its own configuration data.

3017   Note: See *Example 12* for an XML example.

3018   8.6.1 SensorConfiguration

3019 configuration for a `Sensor`.

3020   Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

3021 **8.6.1.1 Value Properties of SensorConfiguration**

3022 *Table 104* 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 104: Value Properties of SensorConfiguration
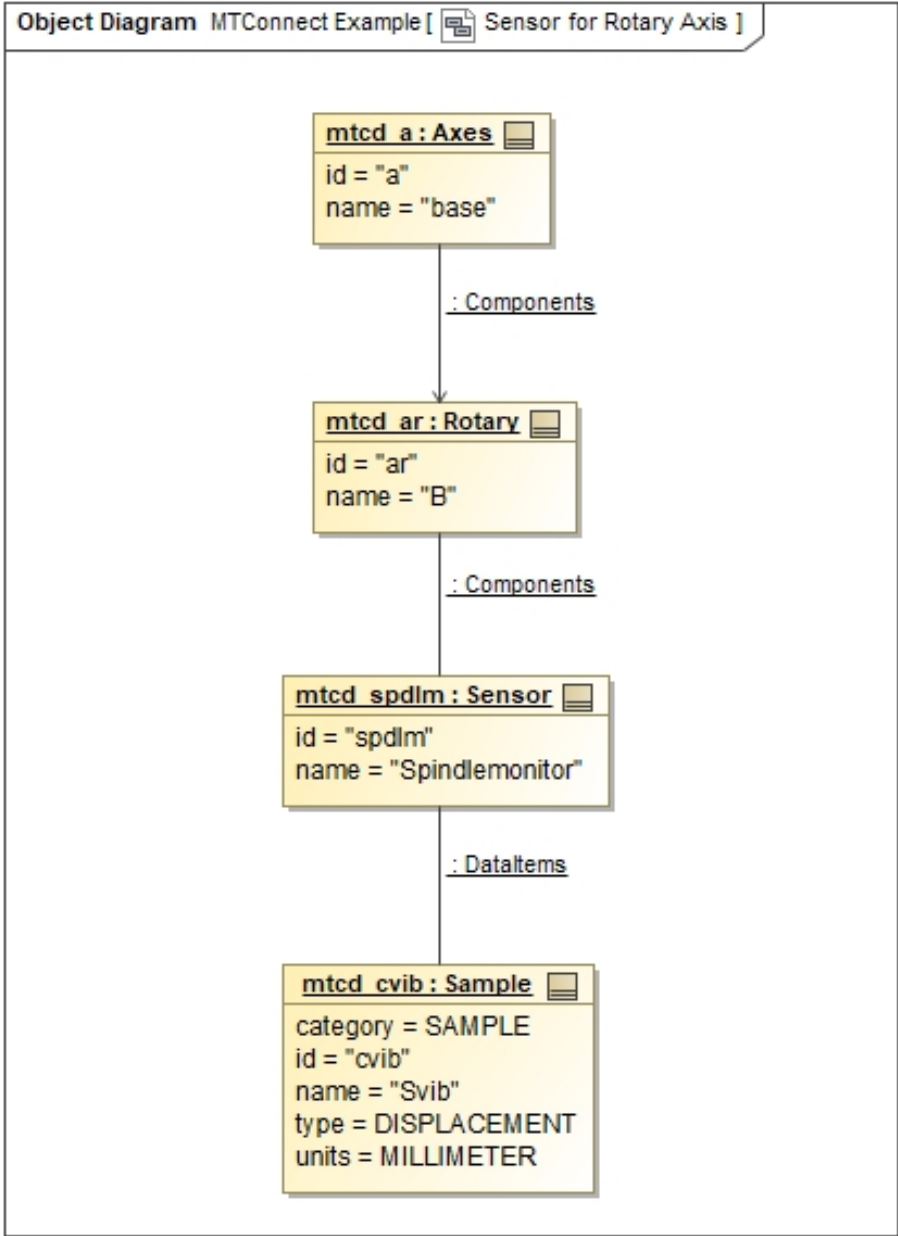
Figure 15: Sensor for Rotary Axis Example
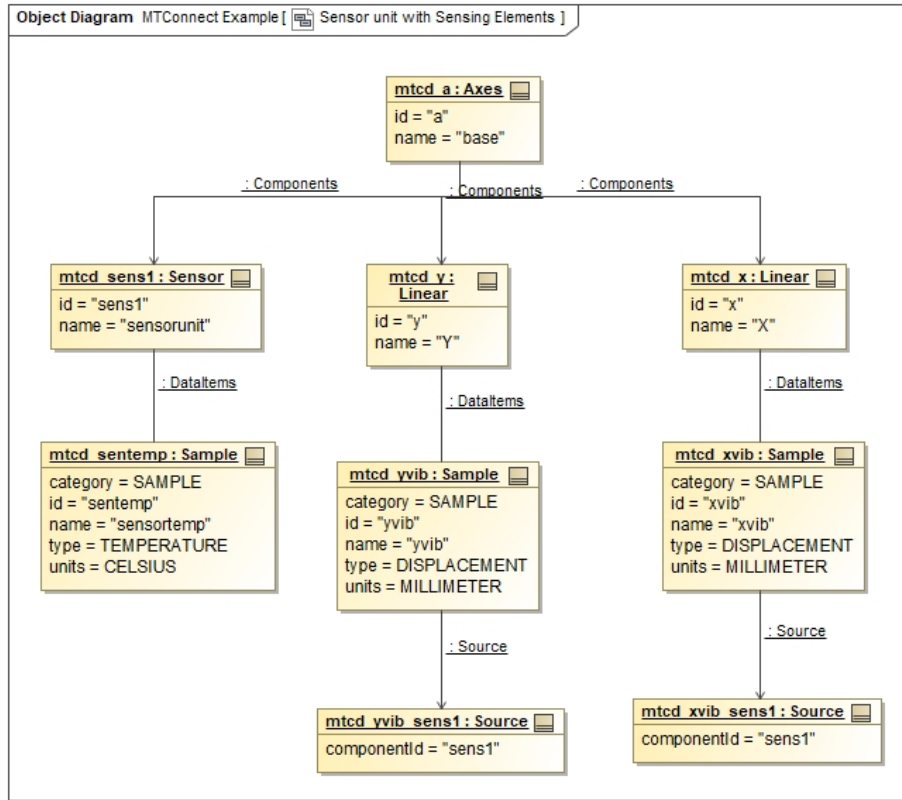
Figure 16: Sensor unit with Sensing Elements Example
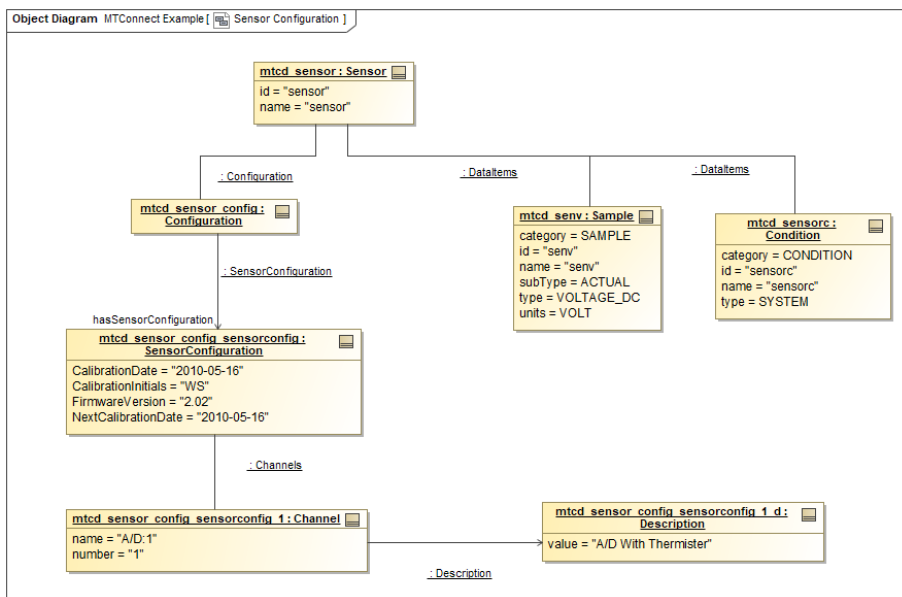

Figure 17: Sensor Configuration Example
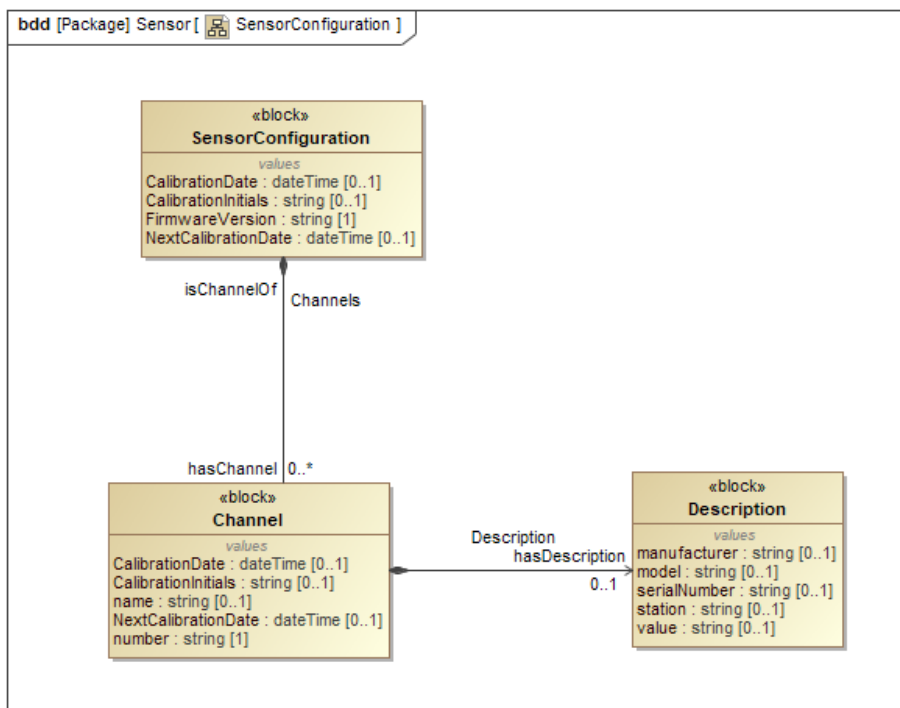
Figure 18: SensorConfiguration

3023 Descriptions for Value Properties of `SensorConfiguration`:

3024 • `CalibrationDate`

3025 Date upon which the sensor unit was last calibrated.

3026 • `CalibrationInitials`

3027 The initials of the person verifying the validity of the calibration data.

3028 • `FirmwareVersion`

3029 Version number for the sensor unit as specified by the manufacturer.

3030 • `NextCalibrationDate`

3031 Date upon which the sensor unit is next scheduled to be calibrated.

3032 **8.6.1.2 Part Properties of SensorConfiguration**

3033 *Table 105* lists the Part Properties of `SensorConfiguration`.

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

Table 105: Part Properties of SensorConfiguration

3034 Descriptions for Part Properties of `SensorConfiguration`:

3035 • `Channel`

3036 *sensing element* of a `Sensor`.

3037 `Channels` groups one or more `Channel` entities. See `Channel`.

3038 **8.6.2 Channel**

3039 *sensing element* of a `Sensor`.

3040 When `Sensor` has multiple *sensing elements*, each *sensing element* is modeled as a
3041 `Channel` for the `Sensor`.

3042 **8.6.2.1 Value Properties of Channel**

3043 *Table 106* 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 106: Value Properties of Channel

3044 Descriptions for Value Properties of `Channel`:

3045 • `CalibrationDate`

3046 Date upon which the sensor unit was last calibrated to the sensor element.

3047 • `CalibrationInitials`

3048 The initials of the person verifying the validity of the calibration data.

3049 • `name`

3050 name of the specific *sensing element*.

3051 • `NextCalibrationDate`

3052 Date upon which the sensor element is next scheduled to be calibrated with the
3053 sensor unit.

3054 • `number`

3055 unique identifier that will only refer to a specific *sensing element*.

3056 **8.6.2.2 Part Properties of Channel**

3057 *Table 107* lists the Part Properties of `Channel`.

Part Property name	Multiplicity
Description	0..1

Table 107: Part Properties of Channel

3058 Descriptions for Part Properties of `Channel`:

3059 • `Description`

3060 descriptive content.

3061 See *Section 4.1.2 - Description*.

3062 8.7 SolidModel

3063 This section provides semantic information for the `SolidModel` entity.
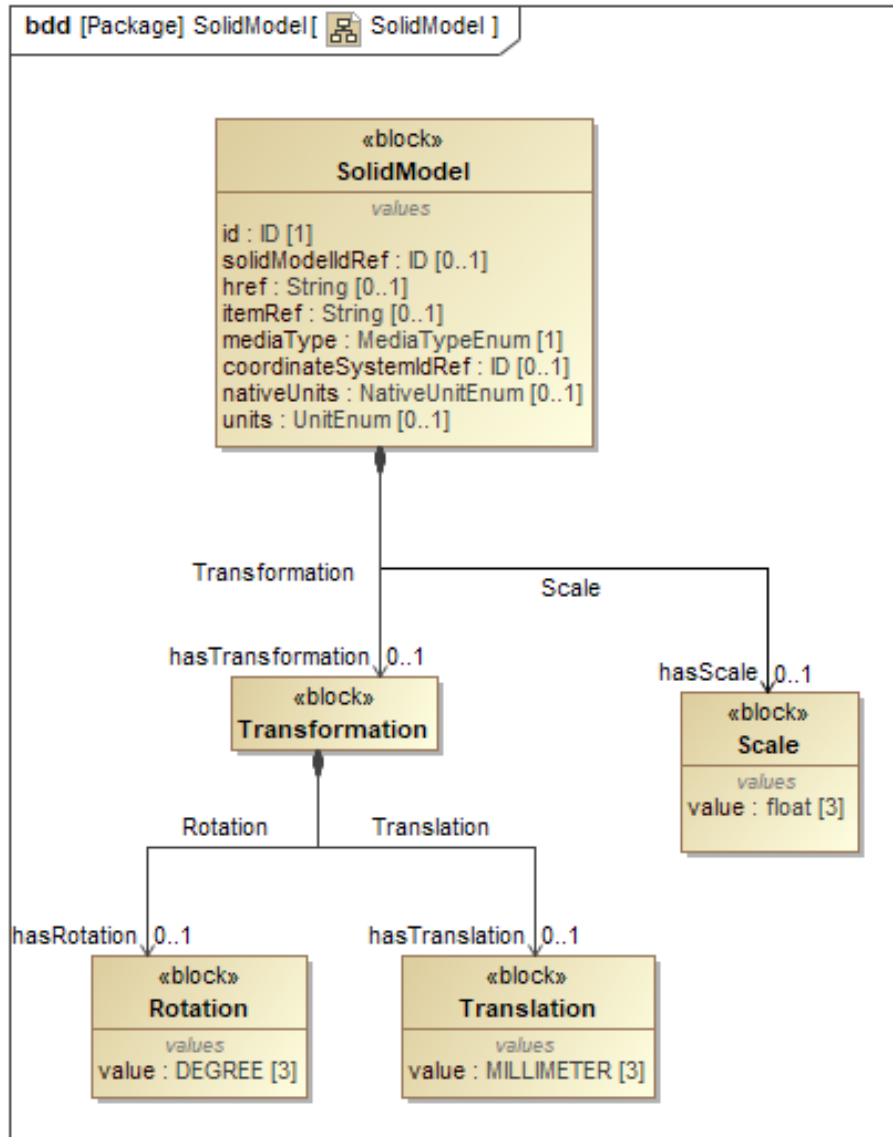
Figure 19: SolidModel

3064 Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

3065 Note: See *Example 7* for the XML representation of the same example.

Figure 20: SolidModel Example

3066 **8.7.1 SolidModel**

3067 references to a file with the three-dimensional geometry of the `Component` or `Compo-`
3068 `sition`.

3069 The geometry **MAY** have a transformation and a scale to position the `Component` with
3070 respect to the other `Components`. A geometry file can contain a set of assembled items,
3071 in this case, the `SolidModel` reference the `id, SolidModel` of the assembly model
3072 file and the specific item within that file.

3073 The `SolidModel` **MAY** provide a translation, rotation, and scale to correctly place it
3074 relative to the other geometries in the machine. If the `Component` can move and has
3075 a `Motion Configuration`, the `SolidModel` will move when the `Component` or
3076 `Composition` moves.

3077 Either an `href, SolidModel` or a `modelIdRef, SolidModel` and an `itemRef, SolidModel`
3078 **MUST** be specified.

3079 **8.7.1.1 Value Properties of SolidModel**

3080 *Table 108* 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 108: Value Properties of SolidModel

3081 Descriptions for Value Properties of `SolidModel`:

3082 • `id`

3083 unique identifier for this element.

3084 • `solidModelIdRef`

3085 associated model file if an item reference is used.

3086 • `href`

3087 URL giving the location of the `SolidModel`.

3088 If not present, the model referenced in the `solidModelIdRef,SolidModel` is
3089 used.

3090 `href,SolidModel` is of type `xlink:href` from the W3C XLink specification.

3091 • `itemRef`

3092 reference to the item within the model within the related geometry. A `solidMod-`
3093 `elIdRef,SolidModel` **MUST** be given.

3094   Note: `Item` defined in ASME Y14.100 - A nonspecific term used to
3095   denote any unit or product, including materials, parts, assemblies, equip-
3096   ment, accessories, and computer software.

3097 • `mediaType`

3098 format of the referenced document.

3099 `MediaTypeEnum` Enumeration:

3100 – 3DS
3101 Autodesk file format.

3102 – ACIS
3103 Dassault file format.

3104 – COLLADA
3105 ISO 17506.

3106 – GDML
3107 Geometry Description Markup Language.

3108 – IGES
3109 Initial Graphics Exchange Specification.

3110 – OBJ
3111 Wavefront OBJ file format.

3112 – STEP
3113 ISO 10303 STEP AP203 or AP242 format.

3114 – STL
3115 STereoLithography file format.

3116 – X_T
3117 Parasolid XT Siemens data interchange format.

3118 • coordinateSystemIdRef
3119 reference to the coordinate system for this SolidModel.

3120 • nativeUnits
3121 same as DataItem nativeUnits. See *Section 6.1.1 - DataItem*.
3122 The value of nativeUnits **MUST** be one of the NativeUnitEnum enumera-
3123 tion.

3124 • units
3125 same as DataItem units. See *Section 6.1.1 - DataItem*.
3126 The value of units **MUST** be one of the UnitEnum enumeration.

3127 **8.7.1.2   Part Properties of SolidModel**

3128 *Table 109* lists the Part Properties of SolidModel.

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

Table 109: Part Properties of SolidModel

3129 Descriptions for Part Properties of `SolidModel`:

3130   • `Transformation`

3131 process of transforming to the origin position of the coordinate system from a parent
3132 coordinate system using `Translation` and `Rotation`.

3133 See *Section 8.2.3 - Transformation*.

3134   • `Scale`

3135 either a single multiplier applied to all three dimensions or a three space multiplier
3136 given in the X, Y, and Z dimensions in the coordinate system used for the `Solid-`
3137 `Model`.

3138 See *Section 8.7.2 - Scale*.

3139 8.7.2   Scale

3140 either a single multiplier applied to all three dimensions or a three space multiplier given
3141 in the X, Y, and Z dimensions in the coordinate system used for the `SolidModel`.

3142 The value of `Scale` **MUST** be a list of `float` of size 3.

3143 8.8   Specifications

3144 This section provides semantic information for the `Specification` entity.

3145 Note: See *Section B.4 - Configuration Schema Diagrams* for XML schema.

3146 Note: See *Example 8* for the XML representation of the same example.
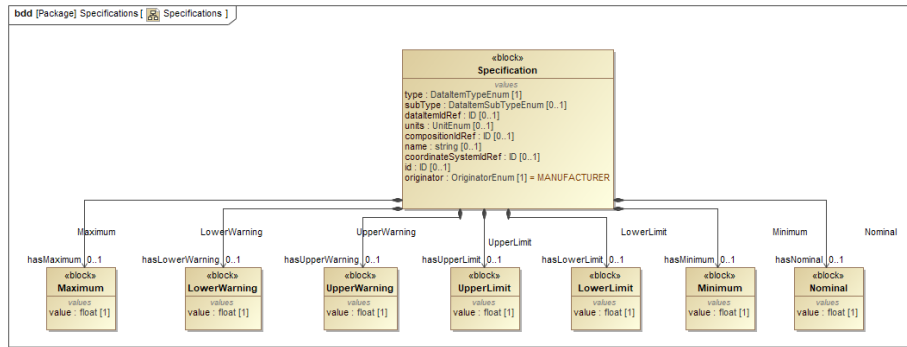
Figure 21: Specifications

3147 ## 8.8.1 Specification

3148 design characteristics for a piece of equipment.

3149 ### 8.8.1.1 Value Properties of Specification

3150 *Table 110* 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 110: Value Properties of Specification
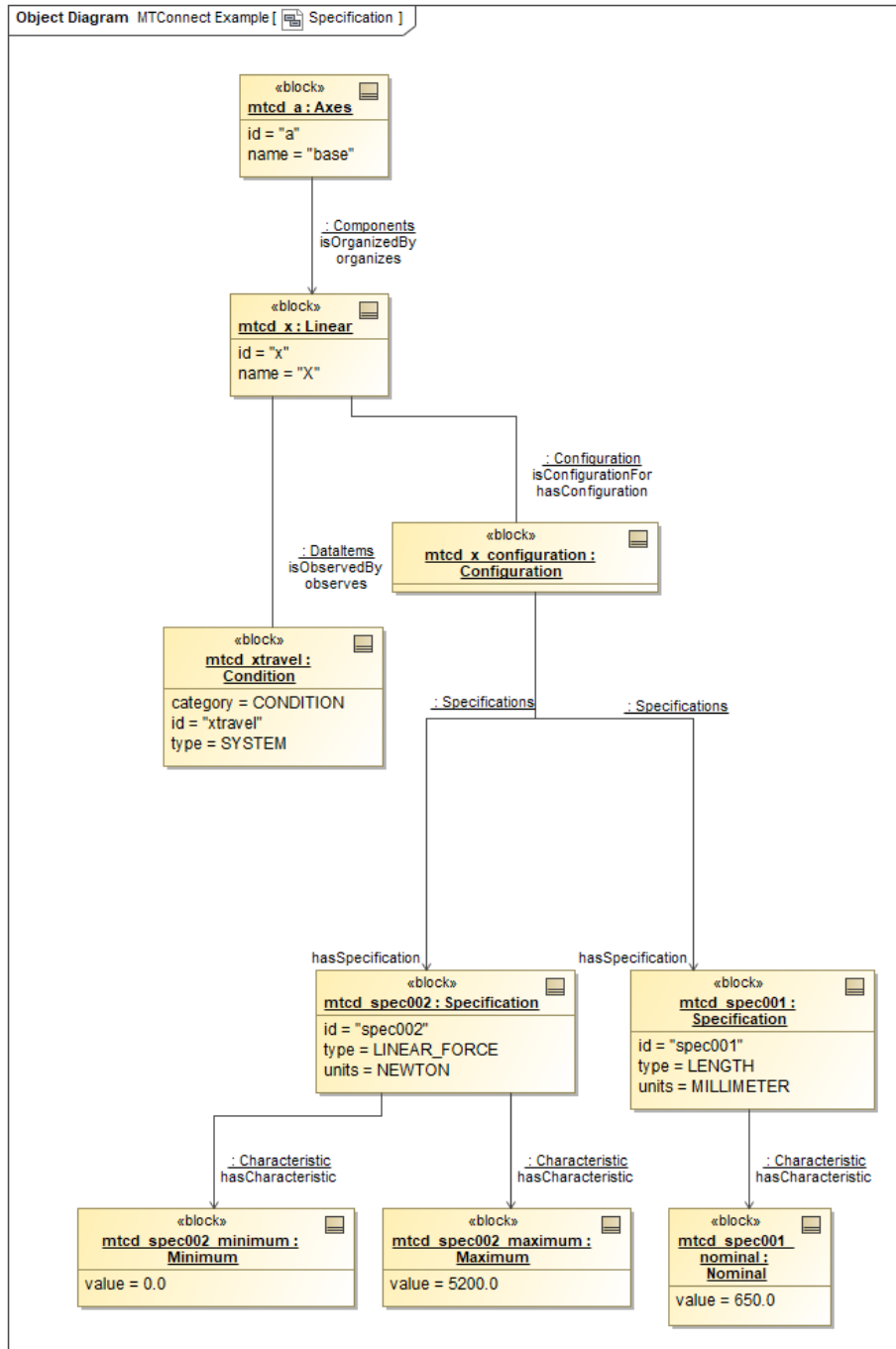
Figure 22: Specification Example

3151 Descriptions for Value Properties of `Specification`:

3152 • `type`

3153 same as `DataItem type`. See *Section 6.5 - DataItem Types*.

3154 The value of `type` **MUST** be one of the `DataItemTypeEnum` enumeration.

3155 • `subType`

3156 same as `DataItem subType`. See *Section 6.1.1 - DataItem*.

3157 The value of `subType` **MUST** be one of the `DataItemSubTypeEnum` enumer-
3158 ation.

3159 • `dataItemIdRef`

3160 reference to the `id` attribute of the `DataItem` associated with this element.

3161 • `units`

3162 same as `DataItem units`. See *Section 6.1.1 - DataItem*.

3163 The value of `units` **MUST** be one of the `UnitEnum` enumeration.

3164 • `compositionIdRef`

3165 reference to the `id` attribute of the `Composition` associated with this element.

3166 • `name`

3167 `name` provides additional meaning and differentiates between `Specification`
3168 elements.

3169 • `coordinateSystemIdRef`

3170 references the `CoordinateSystem` for geometric `Specification` elements.

3171 • `id`

3172 unique identifier for this `Specification`.

3173 • `originator`

3174 reference to the creator of the `Specification`.

3175 `OriginatorEnum` **Enumeration:**

3176 – `MANUFACTURER`
3177 manufacturer of a piece of equipment or `Component`.
3178 – `USER`
3179 owner or implementer of a piece of equipment or `Component`.

3180 **8.8.1.2   Part Properties of Specification**

3181 *Table 111* 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 111: Part Properties of Specification

3182 Descriptions for Part Properties of `Specification`:

3183   • `Maximum`

3184   numeric upper constraint.

3185   • `UpperLimit`

3186   upper conformance boundary for a variable.

3187   Note: immediate concern or action may be required.

3188   • `LowerWarning`

3189   lower boundary indicating increased concern and supervision may be required.

3190   • `LowerLimit`

3191   lower conformance boundary for a variable.

3192   Note: immediate concern or action may be required.

3193   • `UpperWarning`

3194   upper boundary indicating increased concern and supervision may be required.

3195   • `Nominal`

3196   numeric target or expected value.

3197   • `Minimum`

3198   numeric lower constraint.

3199 ## 8.8.2   ProcessSpecification

3200   `Specification` that provides information used to assess the conformance of a variable
3201   to process requirements.

3202 ### 8.8.2.1   Part Properties of ProcessSpecification

3203   *Table 112* lists the Part Properties of `ProcessSpecification`.

Part Property name	Multiplicity
SpecificationLimits	0..1
ControlLimits	0..1
AlarmLimits	0..1

Table 112: Part Properties of ProcessSpecification

3204   Descriptions for Part Properties of `ProcessSpecification`:

3205   • `SpecificationLimits`
3206   set of limits that define a range of values designating acceptable performance for a
3207   variable.
3208   See *Section 8.8.5 - SpecificationLimits*.

3209   • `ControlLimits`
3210   set of limits that is used to indicate whether a process variable is stable and in control.
3211   See *Section 8.8.3 - ControlLimits*.

3212   • `AlarmLimits`
3213   set of limits that is used to trigger warning or alarm indicators.
3214   See *Section 8.8.4 - AlarmLimits*.

3215 ## 8.8.3   ControlLimits

3216   set of limits that is used to indicate whether a process variable is stable and in control.

3217 ### 8.8.3.1   Part Properties of ControlLimits

3218   *Table 113* 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 113: Part Properties of ControlLimits

3219 Descriptions for Part Properties of `ControlLimits`:

3220 • `UpperLimit`

3221 upper conformance boundary for a variable.

3222 Note: immediate concern or action may be required.

3223 • `UpperWarning`

3224 upper boundary indicating increased concern and supervision may be required.

3225 • `LowerWarning`

3226 lower boundary indicating increased concern and supervision may be required.

3227 • `Nominal`

3228 numeric target or expected value.

3229 • `LowerLimit`

3230 lower conformance boundary for a variable.

3231 Note: immediate concern or action may be required.

3232 8.8.4 AlarmLimits

3233 set of limits that is used to trigger warning or alarm indicators.

3234 8.8.4.1 Part Properties of AlarmLimits

3235 *Table 114* lists the Part Properties of `AlarmLimits`.

Part Property name	Multiplicity
UpperLimit	0..1
UpperWarning	0..1
LowerLimit	0..1
LowerWarning	0..1

Table 114: Part Properties of AlarmLimits

3236 Descriptions for Part Properties of `AlarmLimits`:

3237 • `UpperLimit`

3238 upper conformance boundary for a variable.

3239 Note: immediate concern or action may be required.

3240 • `UpperWarning`

3241 upper boundary indicating increased concern and supervision may be required.

3242 • `LowerLimit`

3243 lower conformance boundary for a variable.

3244 Note: immediate concern or action may be required.

3245 • `LowerWarning`

3246 lower boundary indicating increased concern and supervision may be required.

3247 8.8.5   SpecificationLimits

3248 set of limits that define a range of values designating acceptable performance for a variable.

3249 8.8.5.1   Part Properties of SpecificationLimits

3250 *Table 115* lists the Part Properties of `SpecificationLimits`.

Part Property name	Multiplicity
UpperLimit	0..1
Nominal	0..1
LowerLimit	0..1

Table 115: Part Properties of SpecificationLimits

3251 Descriptions for Part Properties of `SpecificationLimits`:

3252 • `UpperLimit`

3253 upper conformance boundary for a variable.

3254 Note: immediate concern or action may be required.

3255 • `Nominal`

3256 numeric target or expected value.

3257 • `LowerLimit`

3258 lower conformance boundary for a variable.

3259 Note: immediate concern or action may be required.

3260 8.8.6 UpperWarning

3261 upper boundary indicating increased concern and supervision may be required.

3262 The value of `UpperWarning` **MUST** be `float`.

3263 8.8.7 UpperLimit

3264 upper conformance boundary for a variable.

3265 Note: immediate concern or action may be required.

3266 The value of `UpperLimit` **MUST** be `float`.

3267 8.8.8 Maximum

3268 numeric upper constraint.

3269 The value of `Maximum` **MUST** be `float`.

3270 8.8.9 LowerLimit

3271 lower conformance boundary for a variable.

3272   Note: immediate concern or action may be required.

3273 The value of `LowerLimit` **MUST** be `float`.

3274 8.8.10 LowerWarning

3275 lower boundary indicating increased concern and supervision may be required.

3276 The value of `LowerWarning` **MUST** be `float`.

3277 8.8.11 Minimum

3278 numeric lower constraint.

3279 The value of `Minimum` **MUST** be `float`.

3280 8.8.12 Nominal

3281 numeric target or expected value.

3282 The value of `Nominal` **MUST** be `float`.

3283 9 Profile

3284 MTConnect Profile is a *profile* that extends the Systems Modeling Language (SysML)
3285 metamodel for the MTConnect domain using additional data types and *stereotypes*.

3286 9.1 DataTypes
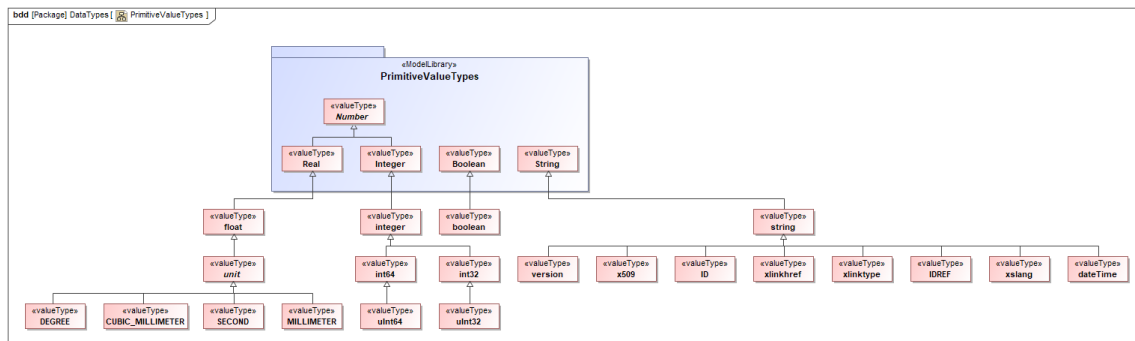
Figure 23: DataTypes

3287 9.1.1 boolean

3288 primitive type.

3289 9.1.2 ID

3290 string that represents an identifier (ID).

3291 9.1.3 string

3292 primitive type.

3293 9.1.4 float

3294 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   IDREF

string that represents a reference to an `ID`.

9.1.11   xlinkhref

string that represents the locator attribute of an XLink element. See `https://www.w3.org/TR/xlink11/`.

3312   9.1.12   x509

3313   string that represents an `x509` data block. *Ref ISO/IEC 9594-8:2020*.

3314   9.1.13   int32

3315   32-bit integer.

3316   9.1.14   int64

3317   64-bit integer.

3318   9.1.15   version

3319   series of four numeric values, separated by a decimal point, representing a *major*, *minor*,
3320   and *revision* number of the MTConnect Standard and the revision number of a specific
3321   *schema*.

3322   9.1.16   uint32

3323   32-bit unsigned integer.

3324   9.1.17   uint64

3325   64-bit unsigned integer.

3326   9.1.18   binary

3327   base-2 numeral system or binary numeral system represented by two digits: "0" and "1".

3328 9.1.19 double

3329 primitive type.

3330 9.2 Stereotypes

3331 9.2.1 organizer

3332 element that *organizes* other elements of a type.

3333 9.2.2 deprecated

3334 element that has been deprecated.

3335 9.2.3 extensible

3336 enumeration that can be extended.

3337 9.2.4 informative

3338 element that is descriptive and non-normative.

3339 9.2.5 valueType

3340 extends SysML `<<ValueType>>` to include `Class` as a value type.

3341 9.2.6 normative

3342 element that has been added to the standard.

3343 **9.2.7   observes**

3344   association in which a *Component* makes *Observations* about an observable *DataItem*.
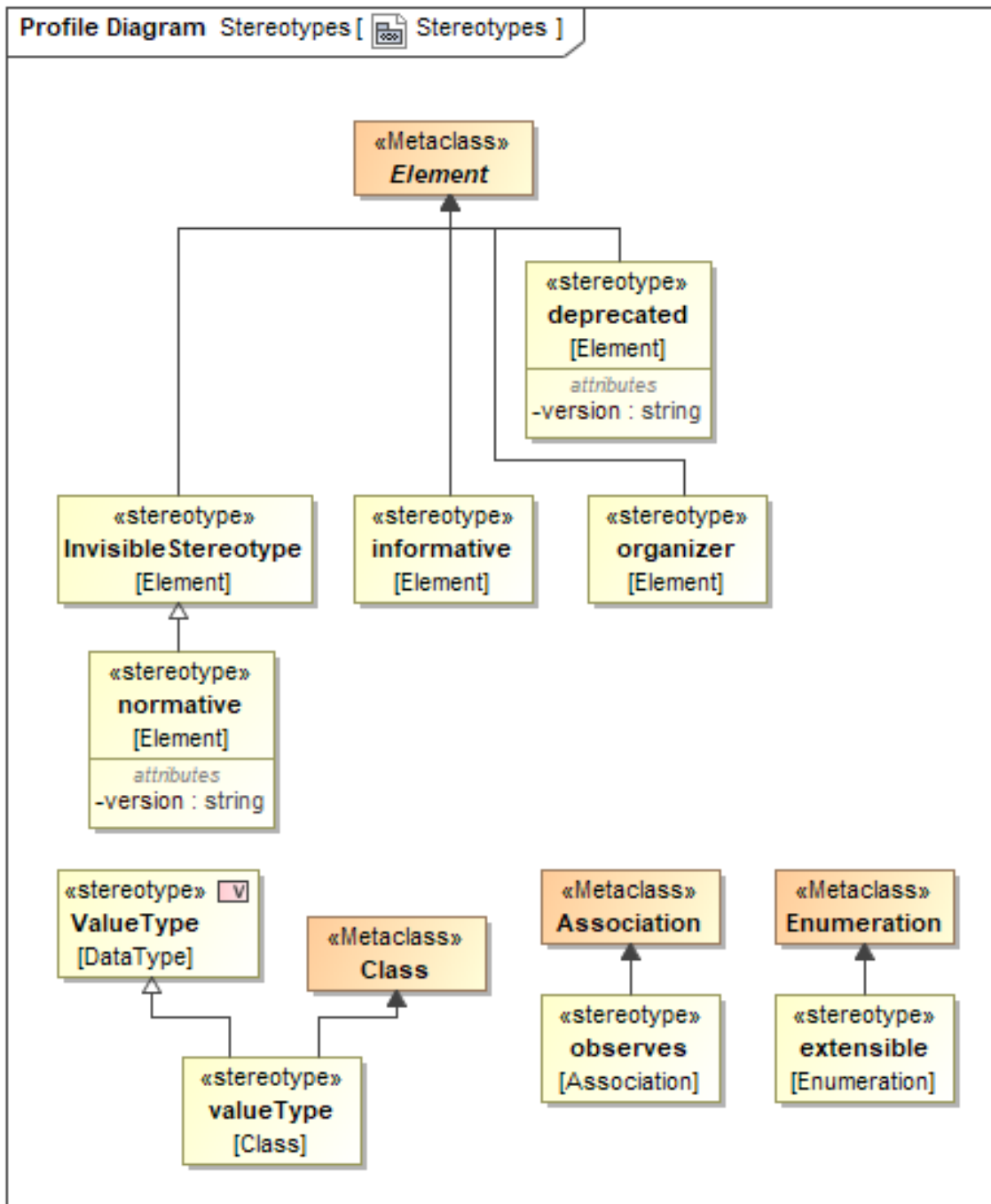
Figure 24: Stereotypes

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.

3396 B   XML Schema Diagrams

3397 See XML schemas for the MTConnect standard here: `https://schemas.mtconnect.`
3398 `org/`.

3399 B.1   Components Schema Diagrams

3400 See `Components` element in `MTConnectDevices` schema.

3401 See `Description` element in `MTConnectDevices` schema.

3402 B.2   DataItems Schema Diagrams

3403 See `DataItems` element in `MTConnectDevices` schema.

3404 See `Source` element in `MTConnectDevices` schema.

3405 See `Constraints` element in `MTConnectDevices` schema.

3406 See `Filter` element in `MTConnectDevices` schema.

3407 See `Definition` element in `MTConnectDevices` schema.

3408 B.3   References Schema Diagrams

3409 See `References` element in `MTConnectDevices` schema.

3410 See `ComponentRef` element in `MTConnectDevices` schema.

3411 See `DataItemRef` element in `MTConnectDevices` schema.

3412 B.4   Configuration Schema Diagrams

3413 See `Configuration` element in `MTConnectDevices` schema.

3414 See `CoordinateSystem` element in `MTConnectDevices` schema.

3415   See `Motion` element in `MTConnectDevices` schema.

3416   See `Relationships` element in `MTConnectDevices` schema.

3417   See `ComponentRelationship` element in `MTConnectDevices` schema.

3418   See `DeviceRelationship` element in `MTConnectDevices` schema.

3419   See `SensorConfiguration` element in `MTConnectDevices` schema.

3420   See `SolidModel` element in `MTConnectDevices` schema.

3421   See `Specifications` element in `MTConnectDevices` schema.

3422   See `ProcessSpecification` element in `MTConnectDevices` schema.

3423 C   XML Examples

3424 C.1   Device Entity Hierarchy Example

Example 1: Device Entity Hierarchy Example

```
3425   1   <Devices>
3426   2     <Device id="d1" name="M12346" uuid="M80104K162N">
3427   3       <Components>
3428   4       <Axes id="a" name="base">
3429   5       <Components>
3430   6         <Linear id="x" name="X"/>
3431   7         <Linear id="y" name="Y"/>
3432   8         <Linear id="z" name="Z"/>
3433   9         <Rotary id="ar" name="B"/>
3434  10         <Linear id="w" name="Z3" nativeName="W"/>
3435  11         <Rotary id="c" name="C"/>
3436  12       </Components>
3437  13       </Axes>
3438  14       <Controller id="cont" name="controller">
3439  15       <Components>
3440  16         <Path id="path1" name="path"/>
3441  17       </Components>
3442  18       </Controller>
3443  19       <Door id="door1" name="door"/>
3444  20       <Systems id="systems" name="systems">
3445  21       <Components>
3446  22         <WorkEnvelope id="we1" name="workenv">
3447  23         <Compositions>
3448  24           <Composition type="WORKPIECE" id="wp"/>
```

```
3449 25          <Composition type="TABLE" id="tbl"/>
3450 26        </Compositions>
3451 27        </WorkEnvelope>
3452 28        <Electric id="elec" name="electric"/>
3453 29        <Hydraulic id="hydraulic" name="hydraulic"/>
3454 30        <Coolant id="coolant" name="coolant"/>
3455 31        <Pneumatic id="pneumatic" name="pneumatic"/>
3456 32        <Lubrication id="lubrication" name="lubrication"/>
3457 33      </Components>
3458 34      </Systems>
3459 35      <Auxiliaries id="Aux" name="auxiliaries">
3460 36      <Components>
3461 37        <Environmental id="room" name="environmental"/>
3462 38      </Components>
3463 39      </Auxiliaries>
3464 40      <Resources id="resources" name="resources">
3465 41      <Components>
3466 42        <Personnel id="personnel" name="personnel"/>
3467 43        <Materials id="materials" name="materials">
3468 44        <Components>
3469 45          <Stock id="procstock" name="stock"/>
3470 46        </Components>
3471 47        <Materials/>
3472 48      </Components>
3473 49      </Resources>
3474 50      </Components>
3475 51    </Device>
3476 52 </Devices>
```

C.2   Component Example

Example 2: Component Example

```
3478  1  <Devices>
3479  2    <Device id="d1" name="M12346" uuid="M80104K162N">
3480  3      <Description manufacturer="Example\textunderscore_Corporation"
3481  4        serialNumber="272237"> Mill w/SMooth-G
3482  5      </Description>
3483  6    <DataItems>
3484  7      <DataItem category="EVENT" id="avail" name="avail" type="
3485     AVAILABILITY"/>
3486  8      <DataItem category="EVENT" id="d1\textunderscore_asset\
3487         textunderscore_chg" name="asset\textunderscore_chg" type="
3488         ASSET\textunderscore_CHANGED"/>
3489  9      <DataItem category="EVENT" id="d1\textunderscore_asset\
3490         textunderscore_rem" name="asset\textunderscore_rem" type="
3491         ASSET\textunderscore_REMOVED"/>
3492 10    </DataItems>
```

```
3493 11        <Components>
3494 12        <Controller id="cont" name="controller">
3495 13          <DataItems>
3496 14          <DataItem category="EVENT" id="estop" name="estop" type="
3497       EMERGENCY\textunderscore_STOP"/>
3498 15          <DataItem category="CONDITION" id="logic\textunderscore_cond"
3499             type="LOGIC\textunderscore_PROGRAM"/>
3500 16          <DataItem category="CONDITION" id="cont\textunderscore_system"
3501              type="SYSTEM"/>
3502 17          <DataItem category="SAMPLE" id="cut\textunderscore_time" type=
3503              "ACCUMULATED\textunderscore_TIME" units="SECOND"/>
3504 18        </DataItems>
3505 19        <Components>
3506 20          <Path id="path1" name="path">
3507 21            <DataItems>
3508 22            <DataItem category="EVENT" id="execution" name="execution"
3509       type="EXECUTION"/>
3510 23            <DataItem category="SAMPLE" id="cspeed" subType="ACTUAL"
3511                type="CUTTING\textunderscore_SPEED" units="MILLIMETER/
3512                SECOND"/>
3513 24            <DataItem category="CONDITION" id="path\textunderscore_
3514                system" type="SYSTEM"/>
3515 25            <DataItem category="EVENT" id="cvars" representation="DATA\
3516                textunderscore_SET" type="VARIABLE"/>
3517 26            <DataItem category="EVENT" id="woffset" representation="
3518                TABLE" type="WORK\textunderscore_OFFSET"/>
3519 27            </DataItems>
3520 28          </Path>
3521 29        </Components>
3522 30        </Controller>
3523 31      </Components>
3524 32      </Device>
3525 33  </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 3: Component with Compositions Example

```
3531 1  <WorkEnvelope id="we1" name="workenv">
3532 2    <DataItems>
3533 3      <DataItem category="CONDITION" id="hardware1" name="hardware1"
3534       type="HARDWARE"/>
3535 4    </DataItems>
```

```
3536  5      <Compositions>
3537  6      <Composition type="WORKPIECE" id="wp"/>
3538  7      <Composition type="TABLE" id="tbl"/>
3539  8      </Compositions>
3540  9  </WorkEnvelope>
```

3541 C.4   CoordinateSystem Example

Example 4: CoordinateSystem Example

```
3542  1   <Devices>
3543  2     <Device id="d1" name="M12346" uuid="M80104K162N">
3544  3       <Description manufacturer="Example\textunderscore_Corporation"
3545  4         serialNumber="272237"> Mill w/SMooth-G
3546  5       </Description>
3547  6       <DataItems>
3548  7         <DataItem id="avail" type="AVAILABILITY" category="EVENT"/>
3549  8         <DataItem category="EVENT" id="d1\textunderscore_asset\
3550            textunderscore_chg" name="asset\textunderscore_chg" type="
3551            ASSET\textunderscore_CHANGED"/>
3552  9         <DataItem category="EVENT" id="d1\textunderscore_asset\
3553            textunderscore_rem" name="asset\textunderscore_rem" type="
3554            ASSET\textunderscore_REMOVED"/>
3555  10      </DataItems>
3556  11      <Configuration>
3557  12        <CoordinateSystems>
3558  13          <CoordinateSystem id="base" type="BASE">
3559  14            <Origin>0 0 0</Origin>
3560  15          </CoordinateSystem>
3561  16          <CoordinateSystem id="machcoord" type="MACHINE" parentIdRef=
3562              "base">
3563  17            <Transformation>
3564  18              <Translation>210 275 1430</Translation>
3565  19              <Rotation>0 0 0</Rotation>
3566  20            </Transformation>
3567  21          </CoordinateSystem>
3568  22          <CoordinateSystem id="workcoord" type="OBJECT" parentIdRef="
3569              machcoord">
3570  23            <Transformation>
3571  24              <Translation>0 0 0</Translation>
3572  25              <Rotation>0 0 0</Rotation>
3573  26            </Transformation>
3574  27          </CoordinateSystem>
3575  28        </CoordinateSystems>
3576  29      </Configuration>
3577  30    </Device>
3578  31  </Devices>
```

3579 C.5 Motion Example

Example 5: Motion Example

```
3580  1  <Devices>
3581  2    <Device id="d1" name="M12346" uuid="M80104K162N">
3582  3      <Description manufacturer="Example\textunderscore_Corporation"
3583  4        serialNumber="272237"> Mill w/SMooth-G
3584  5      </Description>
3585  6      <DataItems>
3586  7        <DataItem id="avail" type="AVAILABILITY" category="EVENT"/>
3587  8        <DataItem category="EVENT" id="d1\textunderscore_asset\
3588           textunderscore_chg" name="asset\textunderscore_chg" type="
3589           ASSET\textunderscore_CHANGED"/>
3590  9        <DataItem category="EVENT" id="d1\textunderscore_asset\
3591           textunderscore_rem" name="asset\textunderscore_rem" type="
3592           ASSET\textunderscore_REMOVED"/>
3593  10     </DataItems>
3594  11     <Configuration>
3595  12       <CoordinateSystems>
3596  13         <CoordinateSystem id="base" type="BASE">
3597  14           <Origin>0 0 0</Origin>
3598  15         </CoordinateSystem>
3599  16         <CoordinateSystem id="machcoord" type="MACHINE" parentIdRef=
3600              "base">
3601  17           <Transformation>
3602  18             <Translation>210 275 1430</Translation>
3603  19             <Rotation>0 0 0</Rotation>
3604  20           </Transformation>
3605  21         </CoordinateSystem>
3606  22         <CoordinateSystem id="workcoord" type="OBJECT" parentIdRef="
3607              machcoord">
3608  23           <Transformation>
3609  24             <Translation>0 0 0</Translation>
3610  25             <Rotation>0 0 0</Rotation>
3611  26           </Transformation>
3612  27         </CoordinateSystem>
3613  28       </CoordinateSystems>
3614  29     </Configuration>
3615  30     <Components>
3616  31       <Axes id="a" name="base">
3617  32         <Components>
3618  33           <Linear id="x" name="X">
3619  34             <Configuration>
3620  35               <Motion id="xmotion" coordinateSystemIdRef="machcoord"
3621          type="PRISMATIC" actuation="DIRECT">
3622  36                 <Transformation>
3623  37                   <Translaton>300 915 590</Translaton>
3624  38                   <Rotation>0 0 0</Rotation>
3625  39                 </Transformation>
3626  40                 <Axis>1.0 0 0</Axis>
```

```
3627  41              </Motion>
3628  42            </Configuration>
3629  43          </Linear>
3630  44          <DataItems>
3631  45            <DataItem id="xtravel" type="SYSTEM" category="CONDITION
3632       ">
3633  46          </DataItems>
3634  47        </Components>
3635  48      </Axes>
3636  49    </Components>
3637  50  </Device>
3638  51  </Devices>
```

C.6 Relationship Example

Example 6: Relationship Example

```
3640   1  <Components>
3641   2    <Axes id="a" name="base">
3642   3      <Components>
3643   4        <Linear id="x" name="X">
3644   5          <Configuration>
3645   6            <Relationships>
3646   7              <ComponentRelationship id="xpar" type="PARENT" idRef="
3647       we1"/>
3648   8            </Relationships>
3649   9          </Configuration>
3650  10        </Linear>
3651  11        <DataItems>
3652  12          <DataItem id="xtravel" type="SYSTEM" category="CONDITION">
3653  13        </DataItems>
3654  14      </Components>
3655  15    </Axes>
3656  16    <Systems id="systems" name="systems">
3657  17      <Components>
3658  18        <WorkEnvelope id="we1" name="workenv">
3659  19          <Compositions>
3660  20            <Composition type="WORKPIECE" id="wp"/>
3661  21            <Composition type="TABLE" id="tbl"/>
3662  22          </Compositions>
3663  23          <DataItems>
3664  24            <DataItem id="we1cond" type="SYSTEM" category="CONDITION">
3665  25          </DataItems>
3666  26        </WorkEnvelope>
3667  27      </Components>
3668  28    </Systems>
3669  29  </Components>
```

3670 C.7 SolidModel Example

Example 7: SolidModel Example

```
3671   1   <Devices>
3672   2     <Device id="d1" name="M12346" uuid="M80104K162N">
3673   3       <Description manufacturer="Example\textunderscore_Corporation"
3674   4         serialNumber="272237"> Mill w/SMooth-G
3675   5       </Description>
3676   6       <DataItems>
3677   7         <DataItem id="avail" type="AVAILABILITY" category="EVENT"/>
3678   8         <DataItem category="EVENT" id="d1\textunderscore_asset\
3679           textunderscore_chg" name="asset\textunderscore_chg" type="
3680           ASSET\textunderscore_CHANGED"/>
3681   9         <DataItem category="EVENT" id="d1\textunderscore_asset\
3682           textunderscore_rem" name="asset\textunderscore_rem" type="
3683           ASSET\textunderscore_REMOVED"/>
3684   10      </DataItems>
3685   11      <Configuration>
3686   12        <CoordinateSystems>
3687   13          <CoordinateSystem id="base" type="BASE">
3688   14            <Origin>0 0 0</Origin>
3689   15          </CoordinateSystem>
3690   16          <CoordinateSystem id="machcoord" type="MACHINE" parentIdRef=
3691             "base">
3692   17            <Transformation>
3693   18              <Translation>210 275 1430</Translation>
3694   19              <Rotation>0 0 0</Rotation>
3695   20            </Transformation>
3696   21          </CoordinateSystem>
3697   22          <CoordinateSystem id="workcoord" type="OBJECT" parentIdRef="
3698             machcoord">
3699   23            <Transformation>
3700   24              <Translation>0 0 0</Translation>
3701   25              <Rotation>0 0 0</Rotation>
3702   26            </Transformation>
3703   27          </CoordinateSystem>
3704   28        </CoordinateSystems>
3705   29      </Configuration>
3706   30      <Components>
3707   31        <Structures id="struct">
3708   32          <Configuration>
3709   33            <SolidModel id="model" mediaType="OBJ" href="/objs/mazak.
3710           obj" coordinateSystemIdRef="base">
3711   34              <Transformation>
3712   35                <Translation>0 860 0</Translation>
3713   36                <Rotation>-90 0 0</Rotation>
3714   37              </Transformation>
3715   38              <Scale>0.001 0.001 0.001</Scale>
3716   39            </SolidModel>
3717   40          </Configuration>
```

```
3718  41            <Components>
3719  42              <Structure id="xaxis" name="X\textunderscore␣AXIS">
3720  43                <Configuration>
3721  44                  <SolidModel id="x\textunderscore␣model" mediaType="OBJ
3722       " itemRef="xaxis" solidModelIdRef="model" coordinateSystemIdRef="
3723       base"/>
3724  45                </Configuration>
3725  46                <DataItems>
3726  47                  <DataItem type="SYSTEM" category="CONDITION" id="
3727       struct\textunderscore␣cond"/>
3728  48                </DataItems>
3729  49              </Structure>
3730  50            </Components>
3731  51          </Structures>
3732  52        </Components>
3733  53      </Device>
3734  54  </Devices>
```

C.8 Specification Example

Example 8: Specification Example

```
3736   1  <Components>
3737   2    <Axes id="a" name="base">
3738   3      <Components>
3739   4        <Linear id="x" name="X">
3740   5          <Configuration>
3741   6            <Specifications>
3742   7              <Specification id="spec001" type="LENGTH" units="
3743       MILLIMETER">
3744   8                <Nominal>650</Nominal>
3745   9              </Specification>
3746  10              <Specification id="spec002" type="LINEAR\textunderscore␣
3747                    FORCE" units="NEWTON">
3748  11                <Maximum>5200</Maximum>
3749  12                <Minimum>0</Minimum>
3750  13              </Specification>
3751  14            </Specifications>
3752  15          </Configuration>
3753  16          <DataItems>
3754  17            <DataItem id="xtravel" type="SYSTEM" category="CONDITION">
3755  18          </DataItems>
3756  19        </Linear>
3757  20      </Components>
3758  21    </Axes>
3759  22  </Components>
```

C.9 Example of sensing element provided as data item associated with a Component

Example 9: Example of sensing element provided as data item associated with a Component

```
1  <Components>
2    <Axes id="a" name="base"
3      <Components>
4        <Rotary id="c" name="C">
5          <DataItems>
6            <DataItem type="TEMPERATURE" id="servotemp4"
7              category="SAMPLE" nativeUnits="CELSIUS"
8              compositionId="Bmotor" units="CELSIUS"/>
9          </DataItems>
10        </Rotary>
11      </Components>
12    </Axes>
13  </Components>
```

C.10 Example of Sensor for rotary axis

Example 10: Example of Sensor for rotary axis

```
1  <Components>
2    <Axes id="a" name="base"
3      <Components>
4        <Rotary id="ar" name="B">
5          <Components>
6            <Sensor id="spdlm" name="Spindlemonitor">
7              <DataItems>
8                <DataItem type="DISPLACEMENT" id="cvib"
9                  category="SAMPLE" name="Svib"
10                 units="MILLIMETER"/>
11             </DataItems>
12           </Sensor >
13           <Components>
14         </Rotary>
15       </Components>
16     </Axes>
17   </Components>
```

C.11 Example of sensor unit with sensing element

Example 11: Example of sensor unit with sensing element

```
3794   1   <Axes id="a" name="base"
3795   2     <Components>
3796   3     <Sensor id="sens1" name="Sensorunit">
3797   4       <DataItems>
3798   5       <DataItem type="TEMPERATURE" id="sentemp"
3799   6         category="SAMPLE" name="Sensortemp"
3800   7         units="CELSIUS"/>
3801   8       </DataItems>
3802   9     </Sensor >
3803  10     <Linear id="x" name="X">
3804  11       <DataItems>
3805  12       <DataItem type="DISPLACEMENT" id="xvib"
3806  13         category="SAMPLE" name="xvib"
3807  14         units="MILLIMETER">
3808  15         <Source componentId="sens1"/>
3809  16       </DataItem>
3810  17       </DataItems>
3811  18     </Rotary>
3812  19     <Linear id="y" name="Y">
3813  20       <DataItems>
3814  21       <DataItem type="DISPLACEMENT" id="yvib"
3815  22         category="SAMPLE" name="yvib"
3816  23         units="MILLIMETER">
3817  24         <Source componentId="sens1"/>
3818  25       </DataItem>
3819  26       </DataItems>
3820  27     </Linear>
3821  28     <Components>
3822  29   </Axes>
```

3823 C.12   Example of configuration data for Sensor

Example 12: Example of configuration data for Sensor

```
3824   1   <Sensor id="sensor" name="sensor">
3825   2     <Configuration>
3826   3       <SensorConfiguration>
3827   4         <FirmwareVersion>2.02</FirmwareVersion>
3828   5         <CalibrationDate>2010-05-16</CalibrationDate>
3829   6         <NextCalibrationDate>2010-05-16</NextCalibrationDate>
3830   7         <CalibrationInitials>WS</CalibrationInitials>
3831   8         <Channels>
3832   9           <Channel number="1" name="A/D:1">
3833  10             <Description>A/D With Thermister</Description>
3834  11           </Channel>
3835  12         </Channels>
3836  13       </SensorConfiguration>
```

```
3837 14    </Configuration>
3838 15    <DataItems>
3839 16      <DataItem category="CONDITION" id="sensorc"
3840 17        name="sensorc" type="SYSTEM" />
3841 18      <DataItem category="SAMPLE" id="senv" name="sensorc"
3842 19        type="VOLTAGE\textunderscore_DC" units="VOLT" subType="ACTUAL"
3843              />
3844 20    </DataItems>
3845 21  </Sensor>
```