NIST GCR 15-991

Recommended practice for the representation of Component catalog data in ISO 10303-210:2014

By

James Stori
SFM Technology

Thomas Thurman
TRThurman Consulting

Allison Barnard Feeney
Kevin Brady
National Institute of Standards and Technology,

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.GCR.991

National Institute of Standards and Technology
U.S. Department of Commerce

NIST GCR 15-991

Recommended practice for the representation of Component catalog data in ISO 10303-210:2014

Prepared for
U.S. Department of Commerce
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-0897

By
James Stori
SFM Technology

Thomas Thurman
TRThurman Consulting

Allison Barnard Feeney
Kevin Brady
National Institute of Standards and Technology,

April 2015

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Acting Under Secretary of Commerce for Standards and Technology and Acting Director

Recommended practice for the representation of component catalog data in ISO 10303-210:2014

Allison Barnard Feeney

Kevin Brady

James Stori

Thomas Thurman

Abstract

This document provides a recommended practice for representing electronic component property data in accordance with ISO 10303-210:2014. It takes into account recent extensions to the EXPRESS schema in that standard and provides illustrations and examples of a typical use case of an integrated circuit amplifier. Example instance data is provided. Detailed examples are providing illustrating construction methodology for composing SI derived units commonly used in electronic component data sheets and catalogs. An example illustrating construction methodology for tabular data is provided. Several detailed examples have been generated demonstrating the population of a number of common characteristics using a variety of both SI base and derived units.

Introduction

The purpose of this document is to provide a recommended practice for the representation of electronic component catalog data in accordance with ISO 10303-210:2014 (STEP AP 210). Catalog data encompasses a wide variety of performance characteristics and specifications, rating limitations, operating recommendations, and ratings provided by component suppliers through product data sheets and catalogs. In current practice, the majority of this information is communicated between component suppliers and their customers through human readable pdf documents containing a combination of tables, data plots, and technical drawings encompassing the necessary functional, physical, and performance characteristics of the device. The scope of this recommended practice is limited to the representation of characteristics that may be represented as individual elements or tables of values, and excludes detailed product model views such as the schematic symbol, footprint, and package, or mechanical outline. The majority of the data requires representing physical quantities with specified units. This document assumes the reader is familiar with ISO 10303 architecture and terminology.

In the remainder of this document, the key Application Reference Model (ARM) Application Objects (AOs), and corresponding Module Interpreted Model (MIM) entities and relationships commonly used in the representation of component catalog data are discussed and illustrated. Several detailed examples have been generated demonstrating the population of a number of common characteristics using a variety of both base and derived units.

Notation

ARM AOs will be denoted with a leading uppercase letter in Arial font (i.e. Functional_unit) while a MIM entity will be displayed in all lowercase notation (i.e. component_functional_unit). In some instance diagrams, the EXPRESS declaration is in all caps (i.e., PRODUCT). In those cases, the figure caption will delineate the context.

Associating parametric data with a part

Figure 1 shows the top-level product structure ARM for a part represented in AP 210. A packaged part represents a combination of some unit of functionality in a physical realization. The Functional_unit_usage_view, Packaged_part, and Package are the three top level usage views typically needed for design application of an electronic component. A usage view is intended to provide sufficient detail to allow the application of a part in an assembly but without sufficient detail to permit reproduction of the part. Each of these usage views is a description of a Product (Functional_product and Part are both subtypes of Product) and may have independent configuration management control applied. It is possible for parametric data to be associated with any Product through a Product_specific_parameter_value_assignment. In the majority of cases, it is appropriate for the catalog data to be associated with the Part of the Packaged_part as it is specific to the physically realized part and is the usual subject of commerce between a supplier and an OEM.

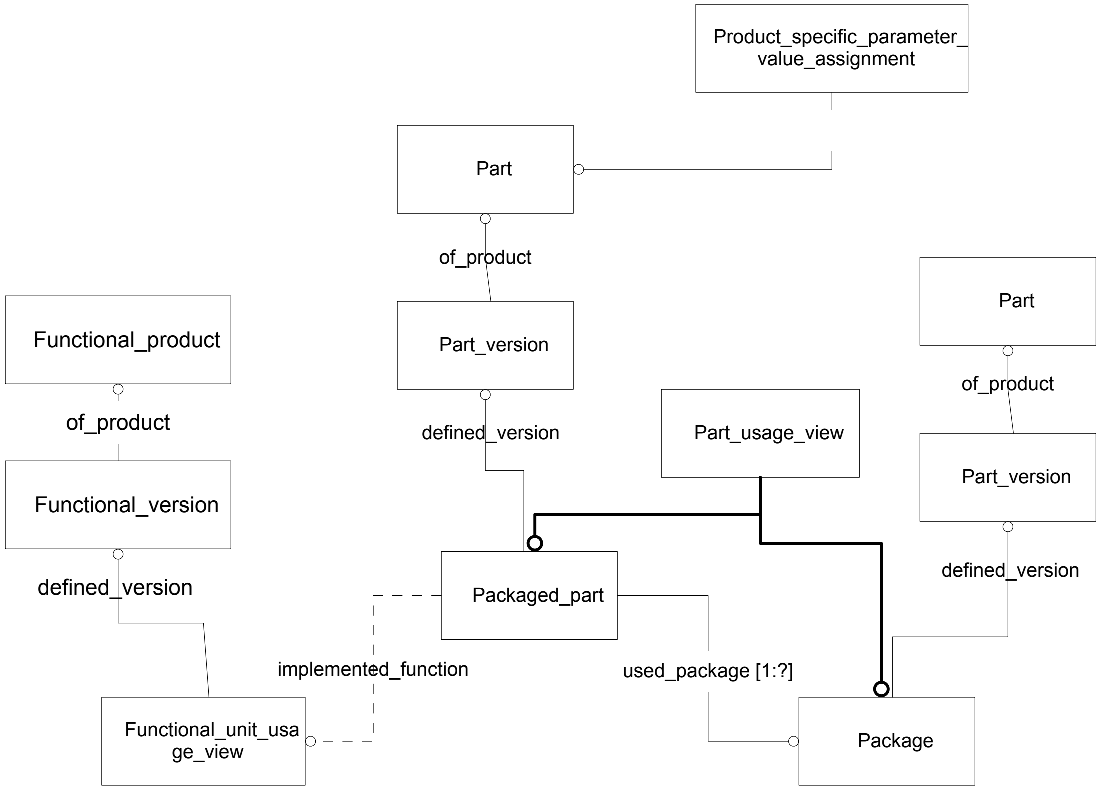

Figure 1. Top-level AOs in the Packaged_part

Figure 2 shows an ARM instance diagram for the same product structure.

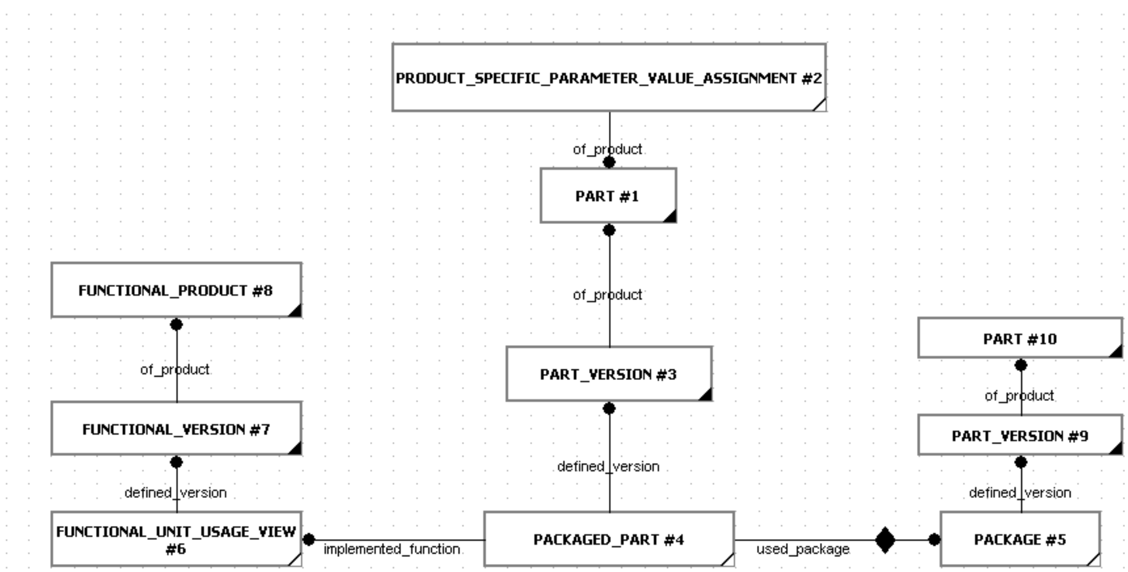

Figure 2. Top-level instance diagram in ARM context

Figure 3 shows the ARM AOs involved in the representation of a parameter, a specific parameter value, and a product. The recommended practice is to represent each characteristic or parameter as a Model_parameter. Examples of such a model parameter would be concepts such as "Supply Voltage," "Storage Temperature Range," "Settling Time," "Bias Current," etc. The MIM entity model_parameter has an id, name, and optional description attribute that may be used to identify and describe the parameter or characteristic. A Parameter_assignment associates a Model_parameter with a specific value. As Parameter_assignment is a subtype of Representation, it may be used to store the represented value itself. The Product_specific_parameter_value_assignment associates a Model_parameter and Parameter_assignment with a specific Product. It is very common for the provided component data to have specified conditions associated with individual values. For example, the bandwidth of an amplifier may vary based on the operating conditions, or the provided typical characteristics may apply at a specific temperature, etc. A complex instance of Parameter_assignment and Material_property_value_representation may be created to associate a Data_environment (representing the environmental conditions) with a value[1]. The Data_environment may specify more than one condition.

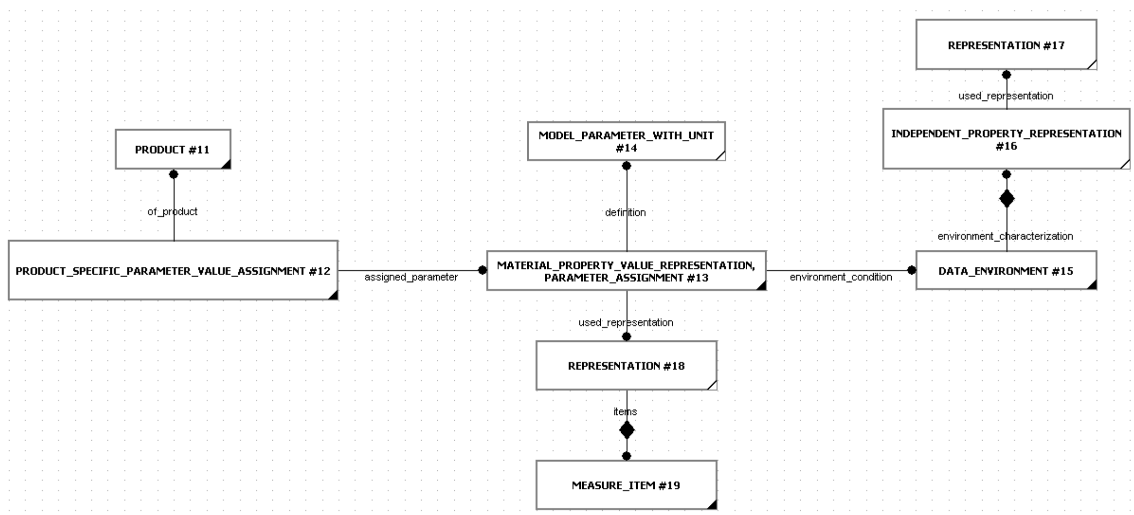

Figure 3. ARM instance diagram illustrating associating a parameter with a product under environmental conditions.

Figure 4 contains a MIM instance diagram of the ARM concepts and relationships related to the parameter assignment and association with a product under environmental

1 This capability was added in edition 3 of AP 210 in order to be consistent with the STEP approach to environmental conditions for test data.

conditions[2]. The environmental conditions is represented by the data_environment MIM entity type and requires the parameter_assignment instance #4 also be an instance of material_property_representation.

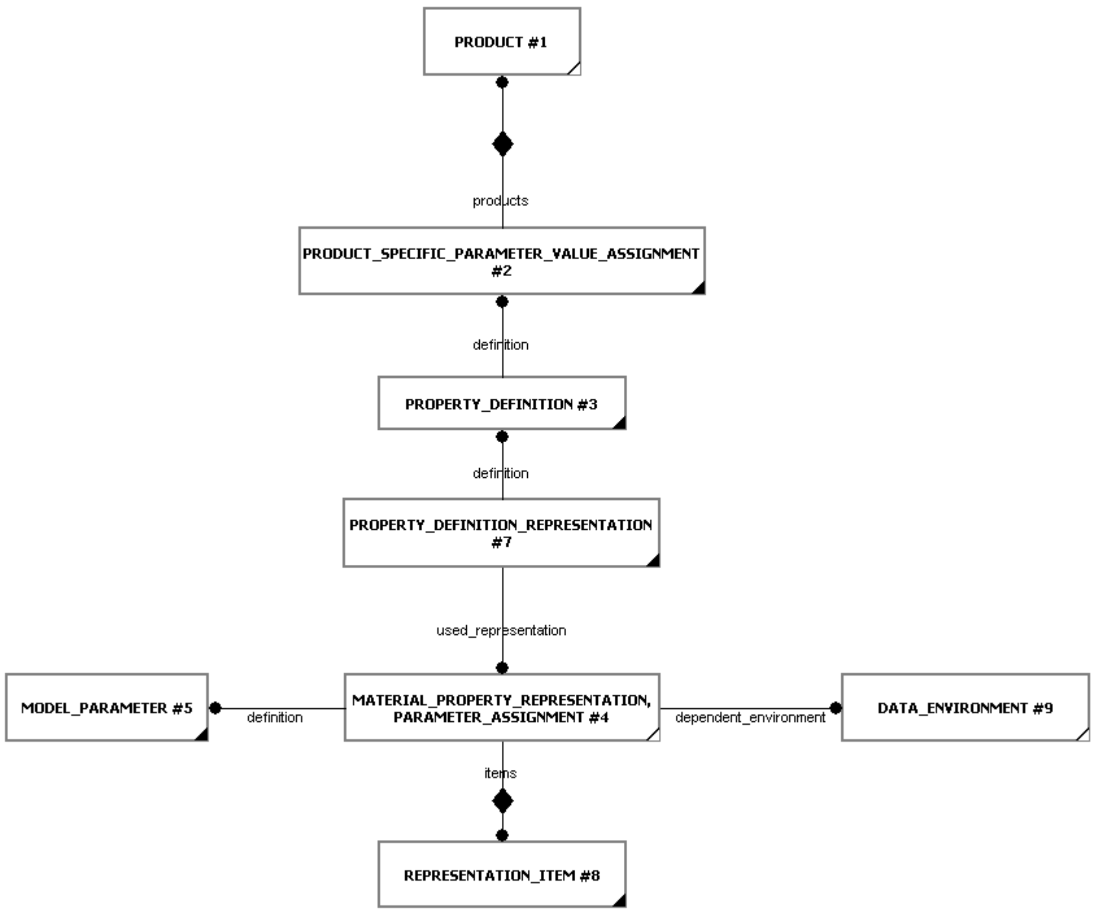

Figure 4. MIM instance diagram associating a parameter with a product under an environmental condition.

A Representation will store the value or values associated with the Model_parameter at the particular conditions specified.

Figure 5 shows the most subtypes of Representation most likely to be employed in the catalog data context. The majority of individual characteristics will likely be represented as a Tolerance_characteristic or a Range_characteristic. A Characteristic_data_table_representation may be used to express a series of values corresponding to a set of given values of an independent parameter. For example, bias current might be provided as a function of temperature or common mode voltage. Either of these mappings could be contained within a table representation. In a data sheet provided by a component manufacturer, such relationships will often be presented in the

2 The ability to support the data structure in Figure 4 is an enhancement provided in the 2014 edition of AP 210.

form of a two-axis plot. It is recommended that such data be provided in table format in the AP 210 model to enable uniform interpretation and representation. For the table data, the plot can also be reconstructed as desired within a receiving application. These three common Representation subtypes will be discussed below in additional detail.

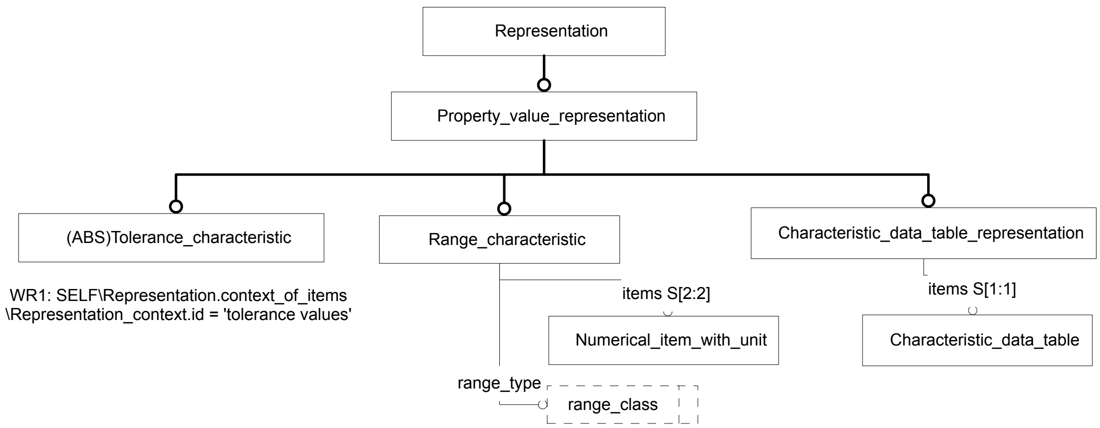

Figure 5. Representation AO type hierarchy.

Representation of a Tolerance_characteristic

Tolerance_characteristic is an abstract supertype, and therefore a specialization must be instantiated.

Figure 6 details some of the common subtypes of Tolerance_characteristic. Many tolerance combinations are both possible and common – in certain cases, only a minimum or maximum may be provided, in other cases minimum and typical values may be available, or often, a combination of minimum, nominal, and maximum values are provided. The Qualified_tolerance_characteristic is intended to represent these common representations.

Many of these tolerance characteristics may be represented in the MIM through the mapping detailed in Figure 7. For uniformity and consistency, it is recommended to use a combination of qualified_representation_item such as described in Figure 7 in preference to some of the more specialized tolerance representations such as Symmetrical_tolerance_characteristic and Plus_minus_tolerance_characteristic.

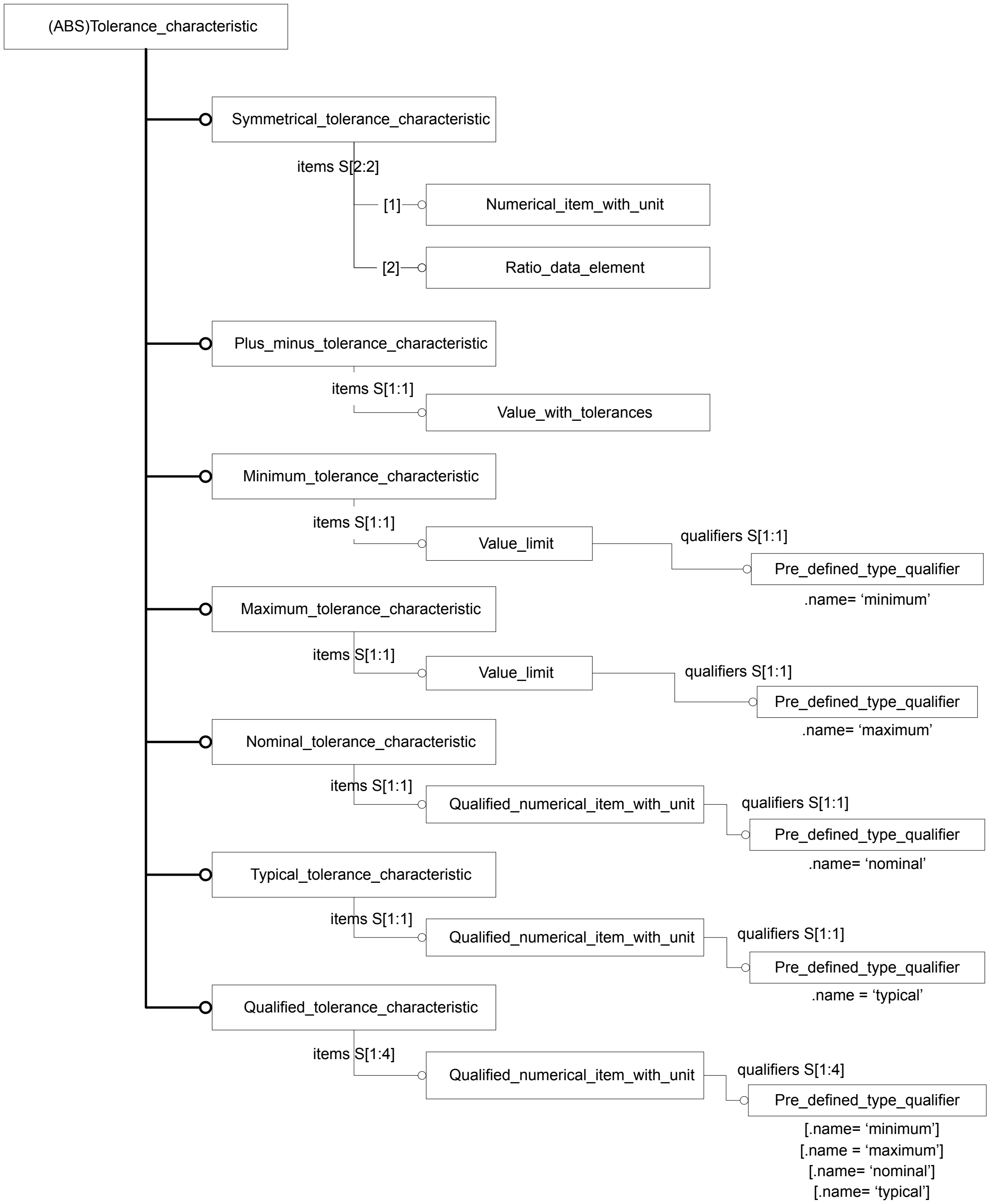

Figure 6. Tolerance_characteristic AO type hierarchy

Representation of a Range_characteristic

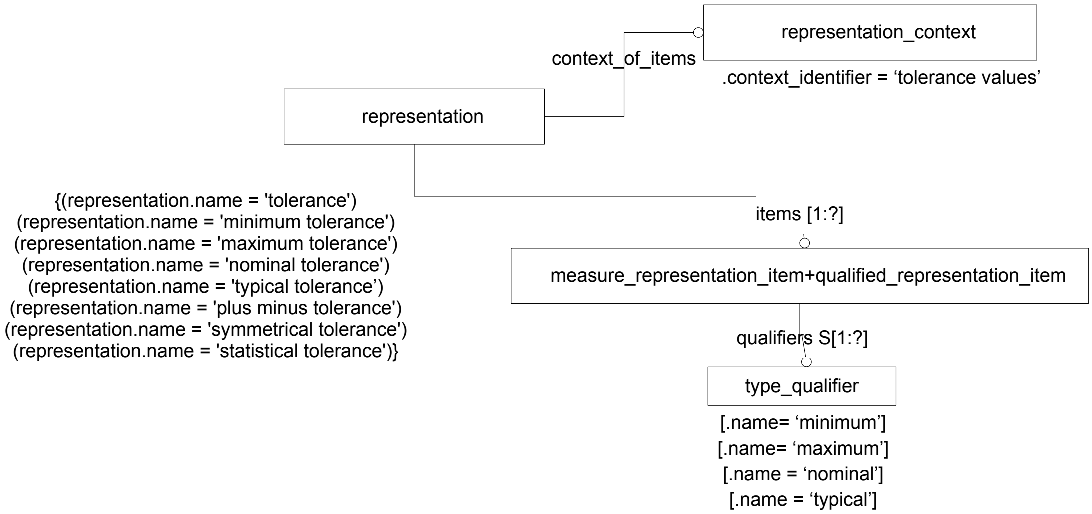

Figure 7. MIM type hierarchy for Tolerance_characteristic AO

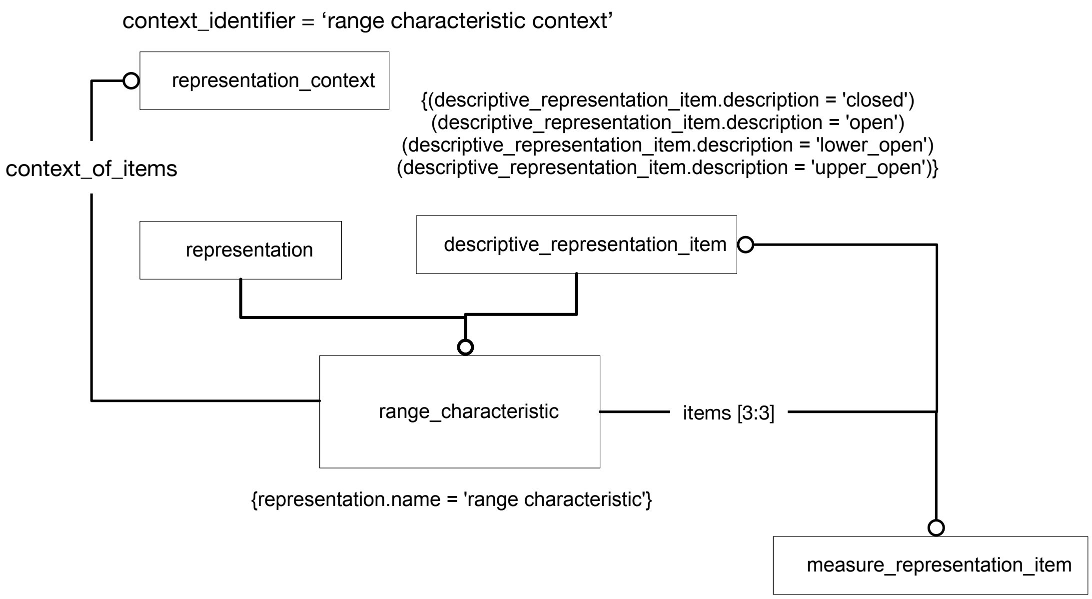

Figure 8. MIM type hierarchy for Range_characteristic AO

A Range_characteristic is intended to describe either an open or closed interval between two measure quantities. The MIM entity range_characteristic inherits from both representation and descriptive_representation_item (see Figure 8). The instance diagram in Figure 9 references the range_characteristic instance (#10) as a member of range_characteristic.items.

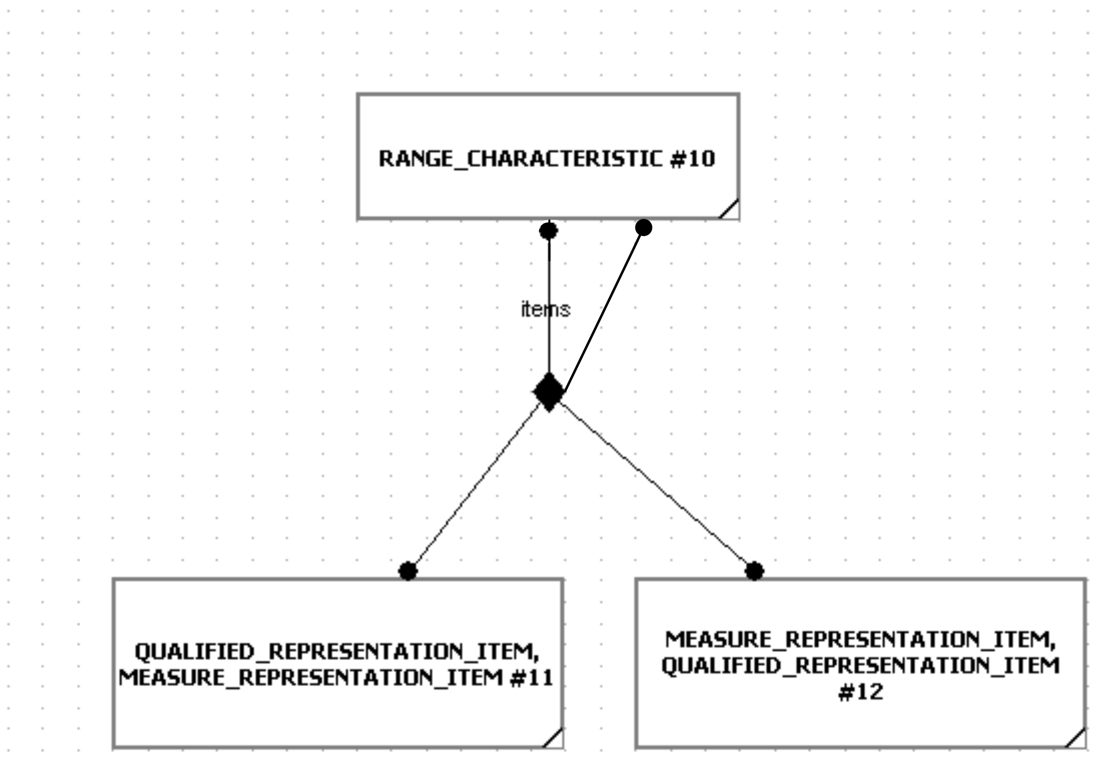

Figure 9. MIM instance data set for Range_characteristic AO

In this alternative mapping a Range_characteristic is similar to a Tolerance_characteristic with an additional descriptive_representation_item containing additional information about the treatment of the interval boundaries (open vs. closed).

Representation of a Numerical_item_with_unit

A measured quantity and an associated unit will be more commonly represented as a Numerical_item_with_unit or it's subtype Qualified_numerical_item_with_unit. The inheritance hierarchy for these AOs is shown in Figure 10.

The MIM mapping for these AOs requires instantiation of a complex entity combining a specific subtype of measure_with_unit (i.e. electric_current_measure_with_unit or time_measure_with_unit) with a measure_representation_item and optionally a qualified_representation_item (in the case of a Qualified_numerical_item_with_unit).

These MIM mappings are detailed in Figure 11.

Figure 12 provides a representative type hierarchy of a Qualified_numerical_item_with_unit for a SI electric potential (i.e. Volt) measure

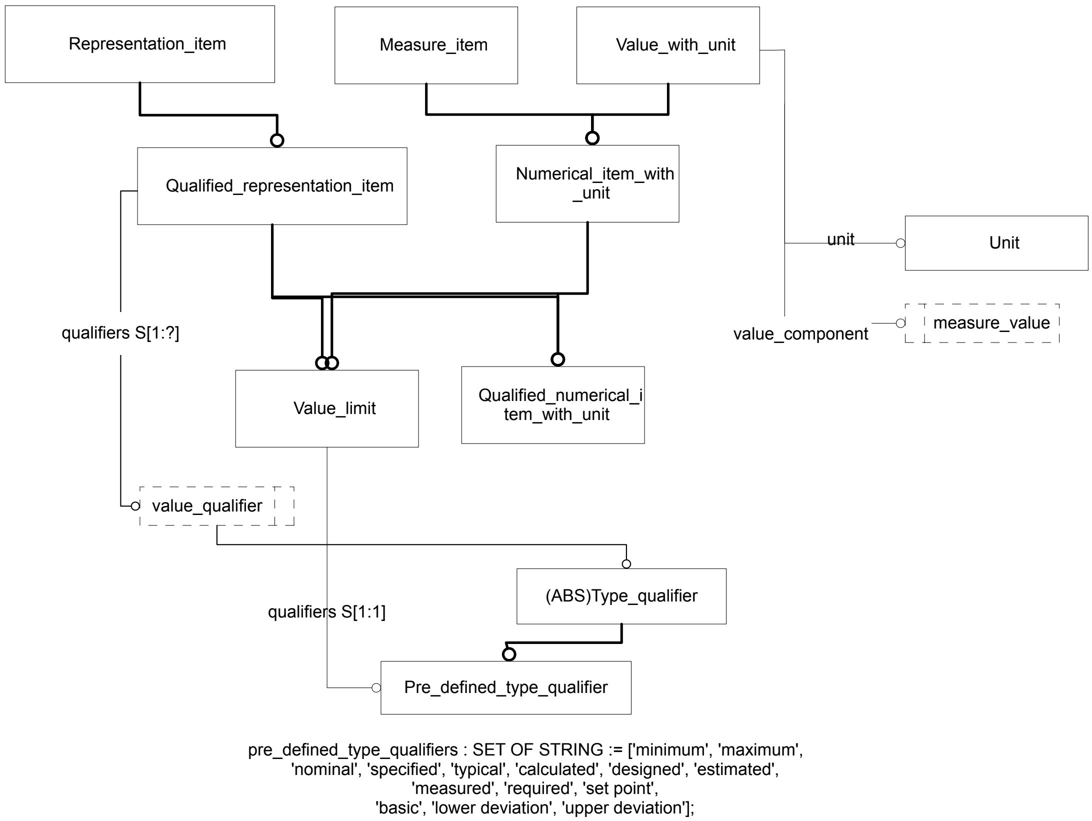

Figure 10. Numerical_item_with_unit AO type hierarchy

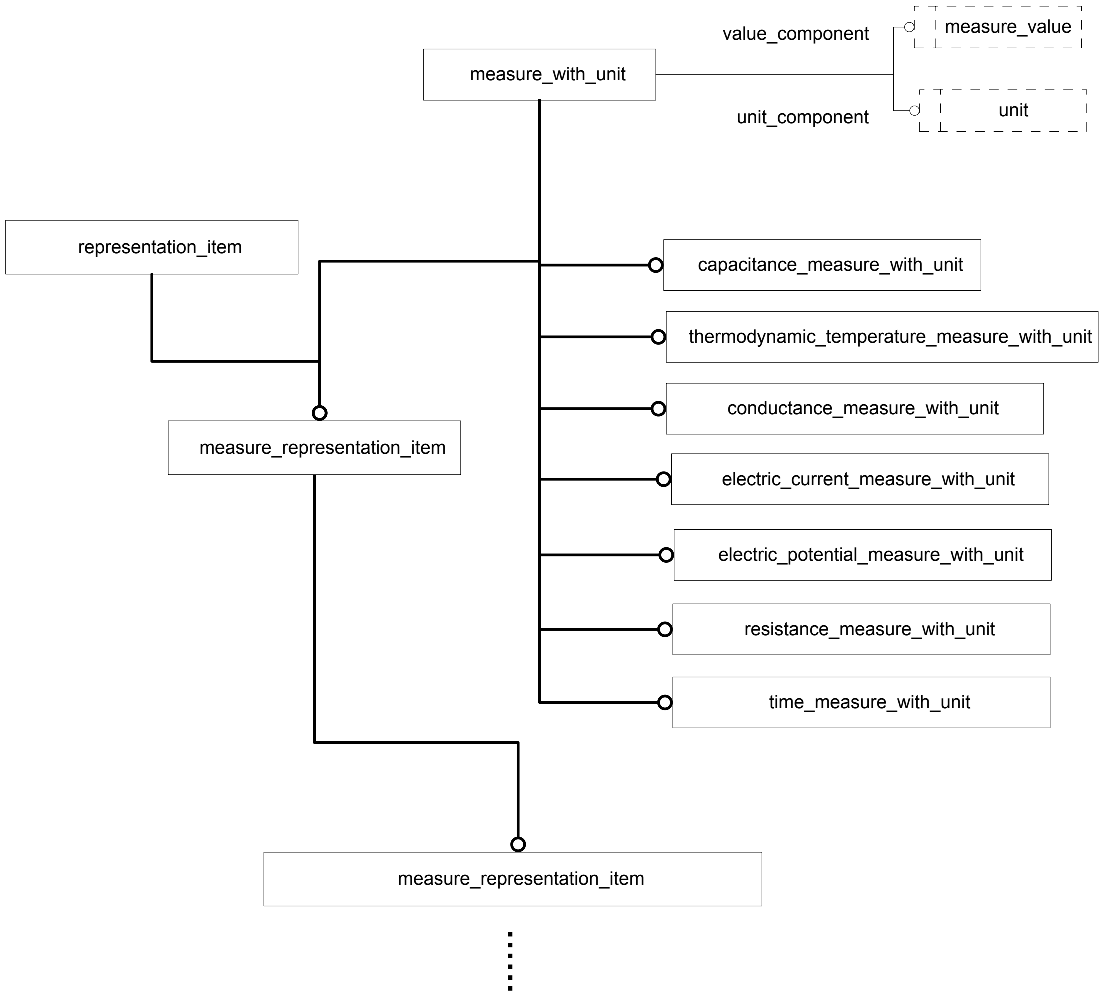

Figure 11. MIM type hierarchy for Numerical_item_with_unit AO

Representation of Unit

The Unit AO enumerates base unit types (Length_unit, Time_unit, Electric_current_unit, etc.) as well as providing a mechanism to define derived units and conversion based units (see Figure 14). As well as defining whether the unit is a SI unit. Area and volume units are correctly handled. Derivation of Fahrenheit temperature scale is supported with a complex model described in part 41 but not in this document. In the MIM, a unit is either a named_unit or a derived_unit – these two entities do not share a common supertype. The base SI unit representations in the MIM will be instantiated as complex entity instances (i.e. si_unit+time_unit) while many of the derived units have explicit SI entity representations (i.e. si_electric_potential_unit).

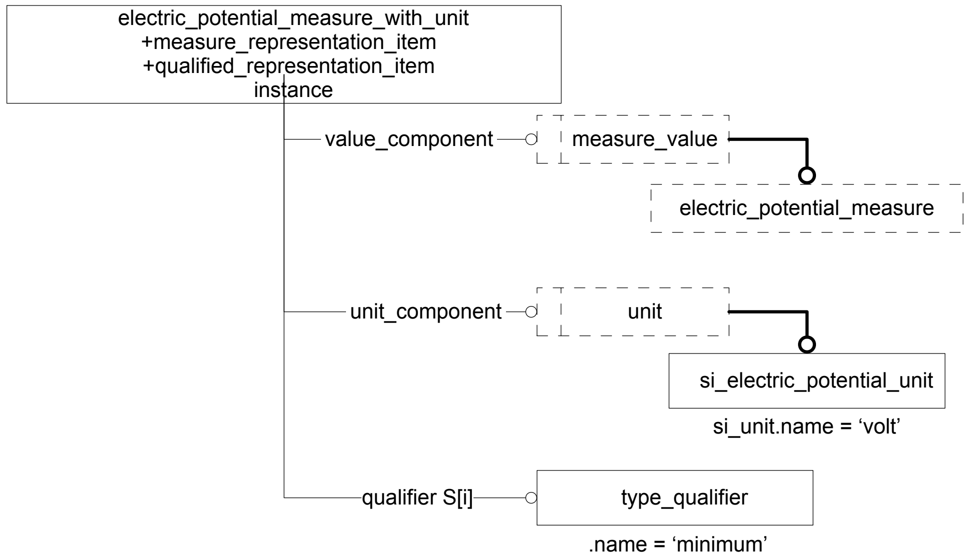

Figure 12. MIM type hierarchy for Qualified_numerical_item_with_unit for Volt

Figure 13 illustrates the MIM mapping of Qualified_numerical_item_with_unit that is a complex instance of of measure_representation_item, qualified_representation_item; it further illustrates an application extension to time_measure_with_unit. It should be noted there is only one instance in the diagram, the more complex instance with time_measure_with_unit.

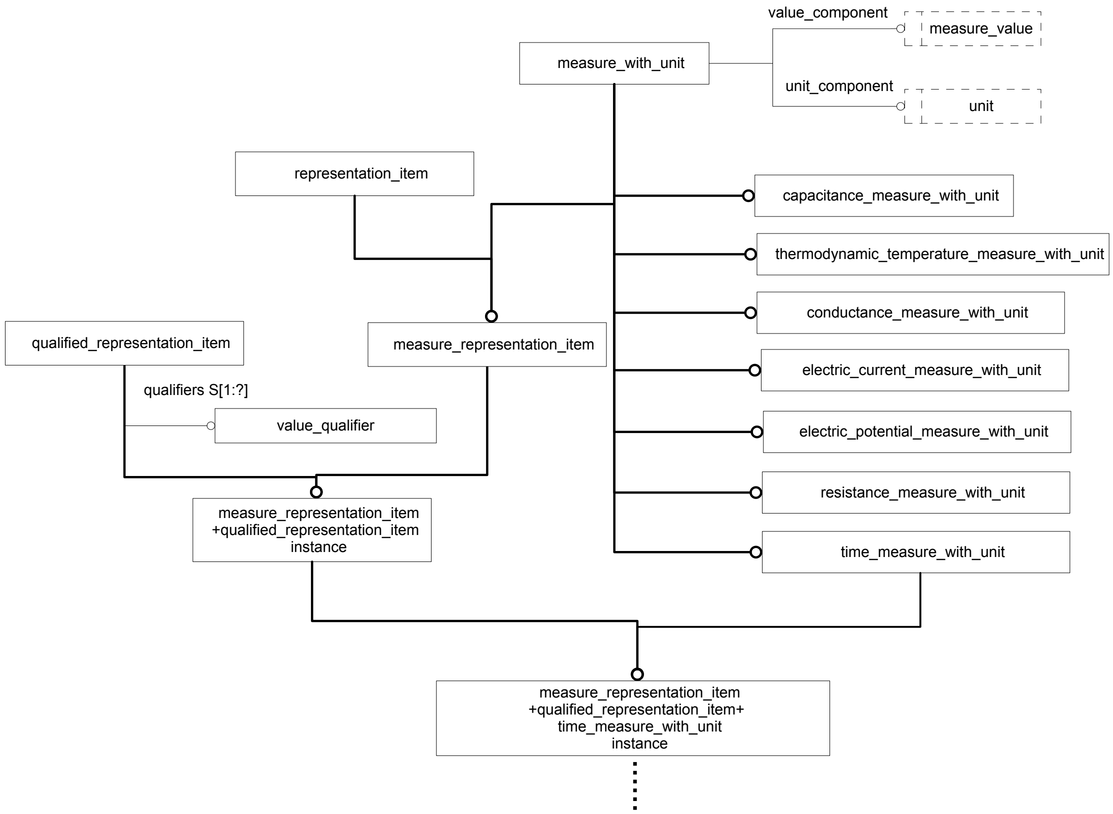

Figure 13. MIM type hierarchy for Qualified_numerical_item_with_unit AO

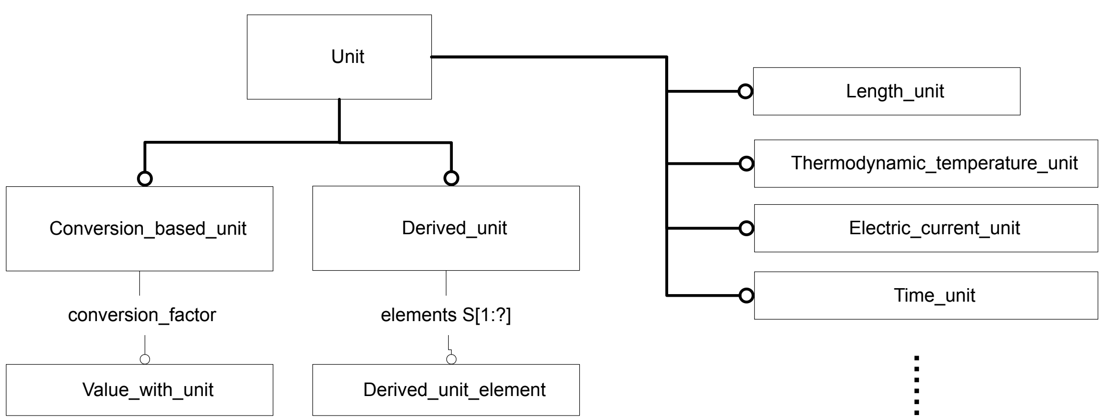

Figure 14. Unit AO type hierarchy

Figure 15 details the inheritance hierarchy and key attributes of named_unit in a combined MIM EXPRESS entity and instance diagram. A complex instance of si_unit + time_unit is shown.

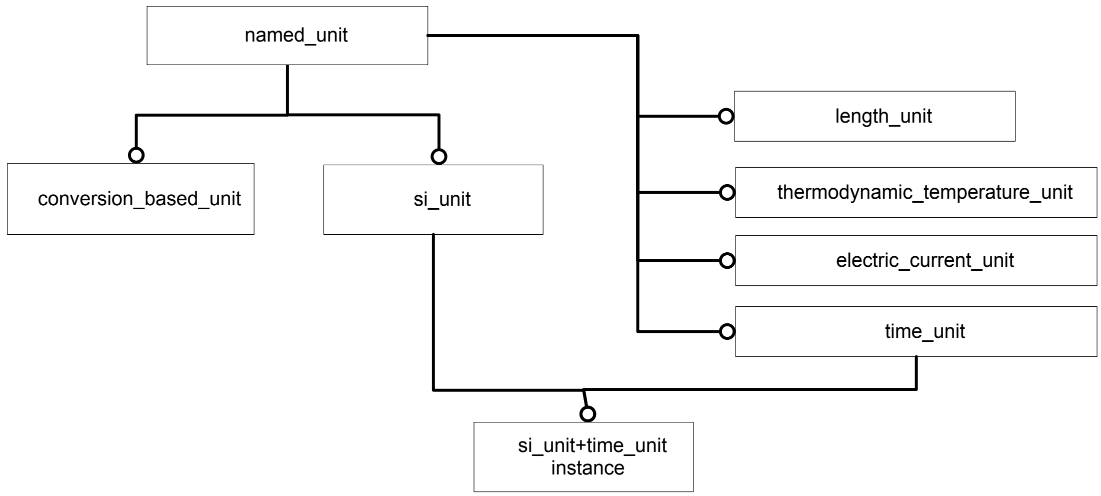

Figure 15. MIM type hierarchy of named_unit.

Figure 16 details the inheritance hierarchy of derived_unit.

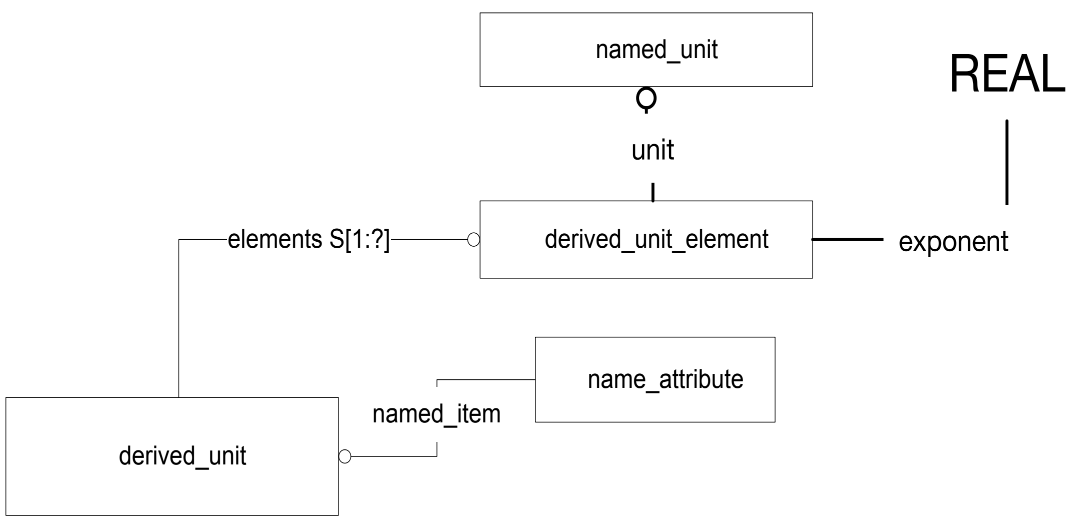

Figure 16. MIM type hierarchy for derived_unit

Several rules and functions in the EXPRESS schema exist to validate and support the use of dimensional_exponents and derived_unit_element for a derived_unit. When representing a derived_unit, there are two possible approaches. The first approach is to define the expression of the derived unit as a function of base units through the population of the appropriate instances of derived_unit_element. A representative example for the case of si_electric_potential_unit is provided in Figure 17. A file snippet if provided below. Each instance of si_unit that is referenced by a derived_unit_element supporting a derive SI unit shall not provide a prefix unless the si_unit being referenced is also a mass_unit, in which case the prefix is required to be .KILO.

part 21 snippet:

```
#24=SI_ELECTRIC_POTENTIAL_UNIT((#25,#27,#29,#31),*,$,.VOLT.);
#25=DERIVED_UNIT_ELEMENT(#26,2.0);
#26=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT($,.METRE.));
#27=DERIVED_UNIT_ELEMENT(#28,1.0);
#28=(MASS_UNIT()NAMED_UNIT(*)SI_UNIT(.KILO.,.GRAM.));
#29=DERIVED_UNIT_ELEMENT(#30,-3.0);
#30=(NAMED_UNIT(*)SI_UNIT($,.SECOND.)TIME_UNIT());
#31=DERIVED_UNIT_ELEMENT(#32,-1.0);
#32=(ELECTRIC_CURRENT_UNIT()NAMED_UNIT(*)SI_UNIT($,.AMPERE.))
```

Because the part 21 snippet is creating a definition for a Volt, only the kilogram is allowed to have a prefix. The instantiation rules for each of the derived SI units using this technique are burdensome when more than a few units are needed in the physical file.

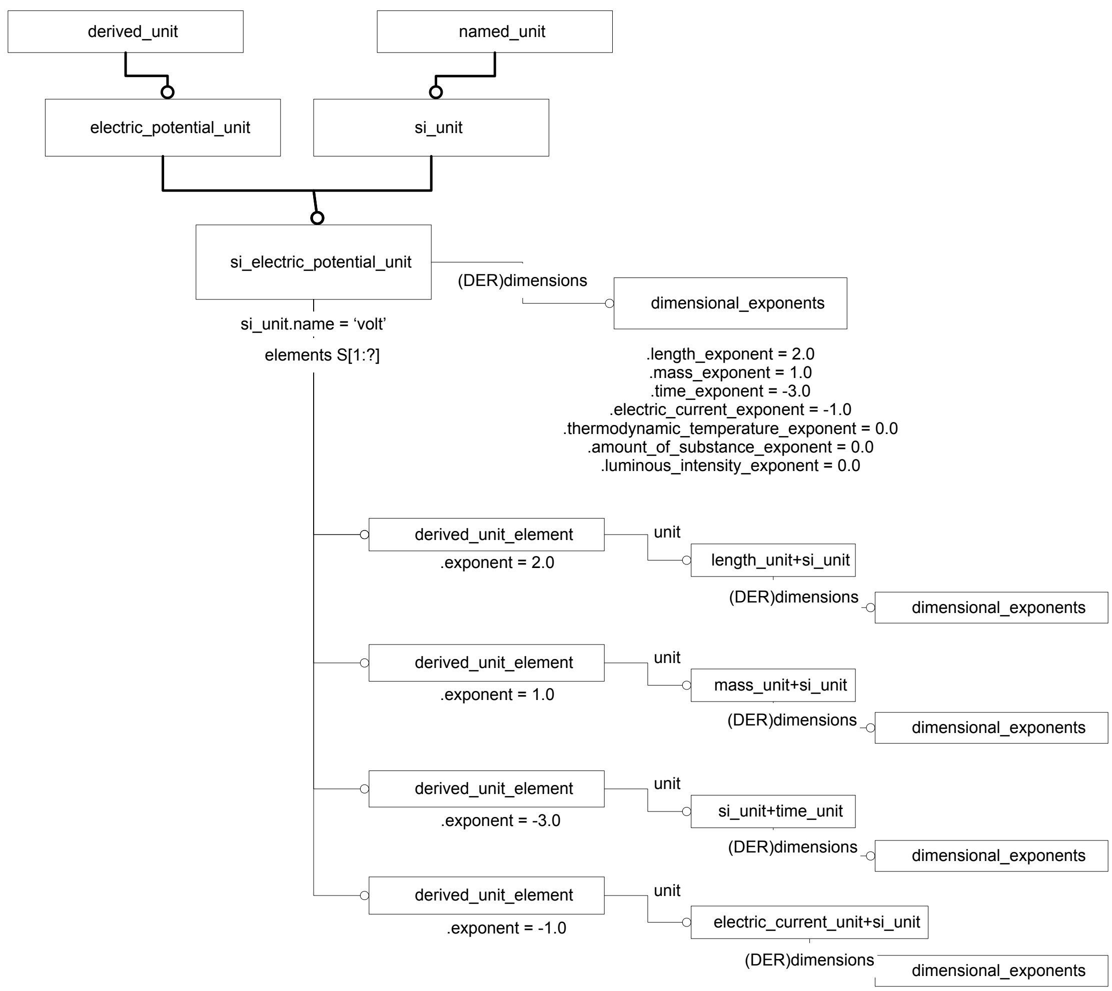

Figure 17. Type hierarchy for SI electric potential unit

The second approach, and that adopted by the CAX-IF (https://cax-if.org) is to establish a chain of derived units once in the physical file. This is easy to follow and verify because it is an incremental approach. Essentially, the CAX-IF authored a set of reference data suitable for use across all participating CAD systems. In the future it is planned to locate a copy of this instance data as reference data at a persistent URI at standards.iso.org to reduce software implementation costs further.

The following (extracted from ISO 10303-41:2014 annex F) is the recommendation for exchange of SI base unit definitions based on the approach of a chain of units.[3]

<u>Base unit reference instances:</u>

```
#4  =(LENGTH_UNIT()NAMED_UNIT(*)            SI_UNIT($,.METRE.));
#14 =(MASS_UNIT()  NAMED_UNIT(*)        SI_UNIT(.KILO.,.GRAM.));4
#24 =(NAMED_UNIT(* SI_UNIT($, .SECOND.)            TIME_UNIT());
#26 =(ELECTRIC_CURRENT_UNIT()NAMED_UNIT(*)SI_UNIT($, .AMPERE.));
#426=(NAMED_UNIT(*)SI_UNIT($,.KELVIN.)
THERMODYNAMIC_TEMPERATURE_UNIT());
#427=(AMOUNT_OF_SUBSTANCE_UNIT() NAMED_UNIT(*) SI_UNIT($, .MOLE.));
#428=(LUMINOUS_INTENSITY_UNIT()  NAMED_UNIT(*) SI_UNIT($, .CANDELA.));
```

SI derived unit exchange should use the derived_unit and unit_elements referencing either a SI base unit or other SI derived units rather than relying directly on dimensional_exponents[5].

The list of entries in si_unit_name in part 41 is not exhaustive. In the case that the name of a derived unit is not included in si_unit_name then an instance of derived_unit (that is not also an instance of si_unit) shall be populated. In that case, the derived_unit.name attribute shall be populated to identify the unit.

---

3 A known issue with the application of a chain of derived units is that the si prefix causes confusion when used to provide scientific notation for a measure value. Example: A 4 megaohm can be represented as:

```
#119=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RESISTANCE_MEASURE(4.0),#121)
     REPRESENTATION_ITEM('')
     RESISTANCE_MEASURE_WITH_UNIT());
#121=SI_RESISTANCE_UNIT((#122,#123,#124,#125),*,.MEGA.,.OHM.);
or as
#119=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RESISTANCE_MEASURE(4.E6),#121)
     REPRESENTATION_ITEM('')
     RESISTANCE_MEASURE_WITH_UNIT());
#121=SI_RESISTANCE_UNIT((#870005,#8700025),*,$,.OHM.);
but
#119=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RESISTANCE_MEASURE(4.0),#121)
     REPRESENTATION_ITEM('')
     RESISTANCE_MEASURE_WITH_UNIT());
#121=SI_RESISTANCE_UNIT((#870005,#8700025),*,.MEGA.,.OHM.);
```

is misleading because a megaohm is not equal to 1 Watt/Amp. The recommendation is to avoid confusion by not providing a prefix for the predefined si derived units.

4 This instance is created to support definition of SI derived units and is the formal definition that the kilogram is the SI unit of mass.

5 TC1 for part41 ed3 addressed the kilogram issue by limiting the application of .KILO. when used to define a unit from the application of .KILO. used as a prefix. It also corrected invalid data structures for area_unit and volume_unit.

In the case that the name of a derived unit is included in si_unit_name, then an instance of the name specific subtype of derived_unit and si_unit shall be populated. In that case, the derived unit name is set equal to the si_unit_name by the schema and any population of derived_unit.name is ignored. The recommendation is to not populate the derived_unit.name attribute.
List of SI derived units whose names are included in si_unit_name:
absorbed_dose_unit,
radioactivity_unit,
capacitance_unit,
dose_equivalent_unit,
electric_charge_unit,
conductance_unit,
electric_potential_unit,
energy_unit,
magnetic_flux_density_unit,
force_unit,
frequency_unit,
illuminance_unit,
inductance_unit,
magnetic_flux_unit,
power_unit,
pressure_unit,
resistance_unit.

The following derived units are included in part 41 but their system of units is unspecified in part 41[6]:
acceleration,
area,
velocity,
volume.

If all derived_unit_element instances reference SI units, then the derived unit is an SI derived unit.

For the case that other units are exchanged (e.g., English engineering) each of the derived_unit_element instances referenced by the derived_unit should be in the same system of units.

A newton is kg-m-sec^{-2}. Part41 provides the ability to explicitly state that force is derived, that the newton is a SI derived unit with a name.

6 Part 41 specifies the fact that it is a derived_unit and specifies the dimensional_exponents values for the unit.

Part 41 _**requires**_ to populate kilogram as the mass unit when newton is defined so as to make the mathematical properties of the data set consistent with SI system of units[7].
Recommended approach:
Definition of Newton:

```
/* establish system of units */
#5=DERIVED_UNIT_ELEMENT(#4,1.0);
#15=DERIVED_UNIT_ELEMENT(#14,1.0);
#25=DERIVED_UNIT_ELEMENT(#24,-2.0);
/* establish newton as SI force unit */
#4161100=SI_FORCE_UNIT((#5,#15,#25),*,$,.NEWTON.);
```

Definition of "pascal":
A Pascal (Pa) is 1 N·m^{-2}.

```
/* establish the multiplication by m-2. */
#550005=DERIVED_UNIT_ELEMENT(#4,-2.0);
/* establish a reference to a newton already defined */
#5500025=DERIVED_UNIT_ELEMENT(#4161100,1.0);
#4161200=SI_PRESSURE_UNIT((#550005,#5500025),*,$,.PASCAL.);
```

Definition of "joule":
A Joule (J) is 1 N·m.

```
/* establish the multiplication by m. */
#650005=DERIVED_UNIT_ELEMENT(#4,1.0);
/* establish a reference to a newton already defined */
#6500025=DERIVED_UNIT_ELEMENT(#4161100,1.0);
#4161300=SI_ENERGY_UNIT((#650005,#6500025),*,$,.JOULE.);
```

Definition of "watt":
A Watt (W) is 1 J·s^{-1}.

```
/* establish the division by sec. */
#750005=DERIVED_UNIT_ELEMENT(#4,-1.0);
/* establish a reference to a joule already defined */
#7500025=DERIVED_UNIT_ELEMENT(#4161300,1.0);
#4161400=SI_POWER_UNIT((#750005,#7500025),*,$,.WATT.);
```

Definition of "coulomb":
A Coulomb (C) is 1 A·s.

```
/* establish system of units */
#8500015=DERIVED_UNIT_ELEMENT(#24,1.0);
#8500025=DERIVED_UNIT_ELEMENT(#26,1.0);
/* establish coulomb as SI electric charge unit */
#1001=SI_ELECTRIC_CHARGE_UNIT((#8500015,#8500025),*,$,.COULOMB.);
```

Definition of "volt":

7 If the `mass_unit` prefix is not provided, even though the newton is declared to be the unit, the numerical instance data declares the unit to be the dyne.

A Volt (V) is 1 W·A^{-1}.

```
/* establish the division by amp. */
#950005=DERIVED_UNIT_ELEMENT(#26,-1.0);
/* establish a reference to a watt already defined */
#9500025=DERIVED_UNIT_ELEMENT(#4161400,1.0);
#1002=SI_ELECTRIC_POTENTIAL_UNIT((#950005,#9500025),*,$,.VOLT.);
```

Definition of "farad":

A Farad (F) is 1 C·V^{-1}.

```
/* establish a reference to a coulomb already defined. */
#860005=DERIVED_UNIT_ELEMENT(#1001,1.0);
/* establish a reference to a volt already defined */
#8600025=DERIVED_UNIT_ELEMENT(#1002,-1.0);
#4161500=SI_CAPACITANCE_UNIT((#860005,#8600025),*,$,.FARAD.);
```

Definition of "ohm":

An Ohm (Ω) is 1 V·A^{-1}.

```
/* establish a reference to a volt already defined. */
#870005=DERIVED_UNIT_ELEMENT(#1001,1.0);
/* establish a reference to amp*/
#8700025=DERIVED_UNIT_ELEMENT(#26,-1.0);
#10099=SI_RESISTANCE_UNIT((#870005,#8700025),*,$,.OHM.);
```

Definition of "siemens":

A Siemens (S) is 1 **Ω**$^{-1}$.

```
/* establish a reference to an ohm already defined. */
#880005=DERIVED_UNIT_ELEMENT(#10099,-1.0);
#100=SI_CONDUCTANCE_UNIT((#880005),*,$,.SIEMENS.);
```

Definition of "weber":

A Weber (Wb) is 1 V·s.

```
/* establish a reference to a volt already defined. */
#890005=DERIVED_UNIT_ELEMENT(#1001,1.0);
/* establish a reference to second*/
#8900025=DERIVED_UNIT_ELEMENT(#24,1.0);
#10023=SI_MAGNETIC_FLUX_UNIT((#890005,#8900025),*,$,.WEBER.);
```

Definition of "tesla":

A Tesla (T) is 1 Wb·m^{-2}.

```
/* establish a reference to a weber already defined. */
#900005=DERIVED_UNIT_ELEMENT(#10023,1.0);
/* establish a reference to metre*/
#9000025=DERIVED_UNIT_ELEMENT(#4,-2.0);
#4161600=SI_MAGNETIC_FLUX_DENSITY_UNIT((#900005,#9000025),*,$,.TESLA.);
```

Definition of "henry":

A Henry (H) is 1 Wb·A^{-1}.

```
/* establish a reference to a weber already defined. */
#910005=DERIVED_UNIT_ELEMENT(#10023,1.0);
/* establish a reference to ampere*/
#9100025=DERIVED_UNIT_ELEMENT(#26,-1.0);
#47000=SI_INDUCTANCE_UNIT((#910005,#9100025),*,$,.HENRY.);
```

Recommendations and examples for non SI unit exchange are available in the following document: http://www.cax-if.de/documents/rec_prac_user_def_attributes_v13.pdf.

Representation of the Characteristic_data_table

In addition to supporting the association of a single value or range with a component characteristic or parameter, a Parameter_assignment can be used to associate a table of values with a Model_parameter. Population of a table of values may be an alternative in certain cases to the use of the explicit conditions in a parameter assignment. For example, if bandwidth specifications are provided at 5 discrete voltages, the implementer has a choice between a table containing 5 rows and two columns as the representation of a single Parameter_assignment, or 5 separate parameter assignments, each with a single voltage value as its condition. Figure 18 contains the MIM mapping of the Characteristic_data_table_representation. Note that the representation instance #15 is the representation that is the map of Characteristic_data_table_representation AO. This is a case where the ARM AO inferred from AIM instance data can be determined only by examining all related attributes (e.g., any representation that has a table_representation in its items attribute and that table_representation has a row_representation_item in its items attribute shall be an implementation of the AO Characteristic_data_table_representation.

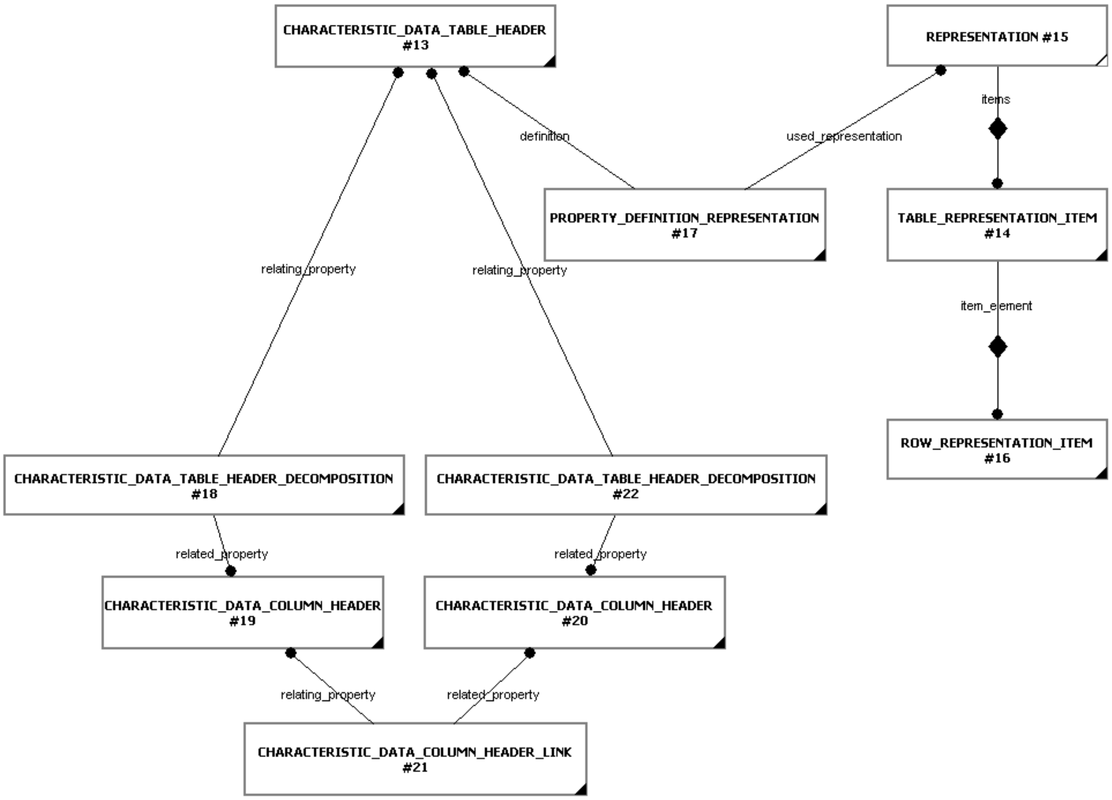

Figure 18. MIM instance diagram of Characteristic_data_table_representation AO

Figure 19 outlines the primary AOs and relationships used in the representation of a Characteristic_data_table_representation. There is one header and a number of rows.

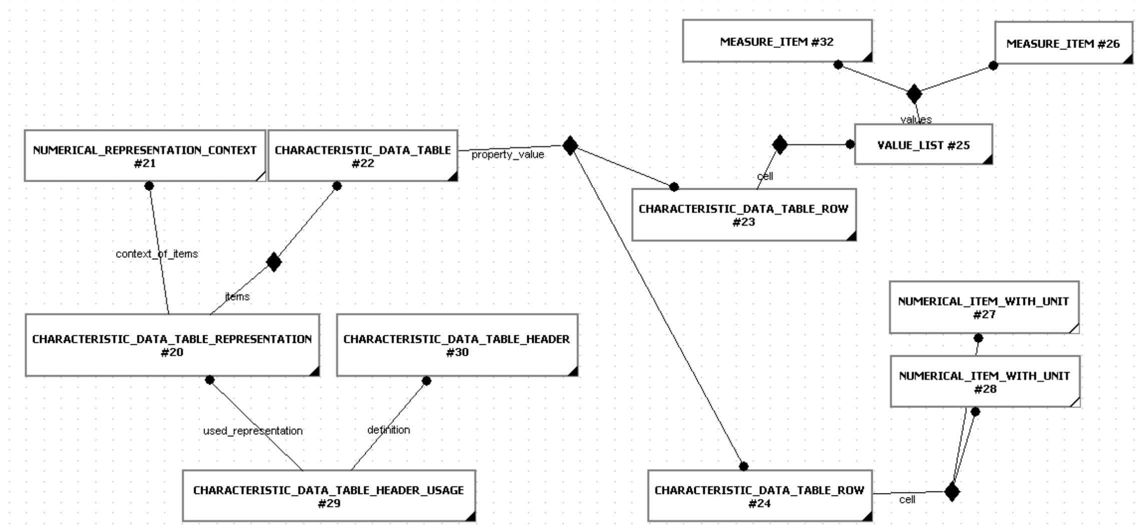

Figure 19. Characteristic_data_table ARM instance diagram

As may be seen in Figure 19, the table representation is constrained to contain exactly one Representation_item, a Characteristic_data_table. The Characteristic_data_table contains a list of rows (Characteristic_data_table_row), each of which, in turn, contains a list of cells. The association between the Characteristic_data_table_representation and the Characteristic_data_table_header is prescribed by the recently[8] added Characteristic_data_table_header_usage AO.

The Characteristic_data_table_header structure is detailed in Figure 20. There is no explicit relationship between the individual column headers and the cells – the relationship must be inferred by the column index position. The structure was created this way to allow use of the column header information in more than one table.

The column headers do not contain a default unit, and it is recommended to populate all cell members explicitly with their own unit, as a Numerical_item_with_unit, as detailed above.

The ARM AOs and corresponding MIM mapping of the Characteristic_data_table_header are detailed in Figures 20 and 23.

8 Characteristic_data_table_header_usage added in AP 210 3rd edition.

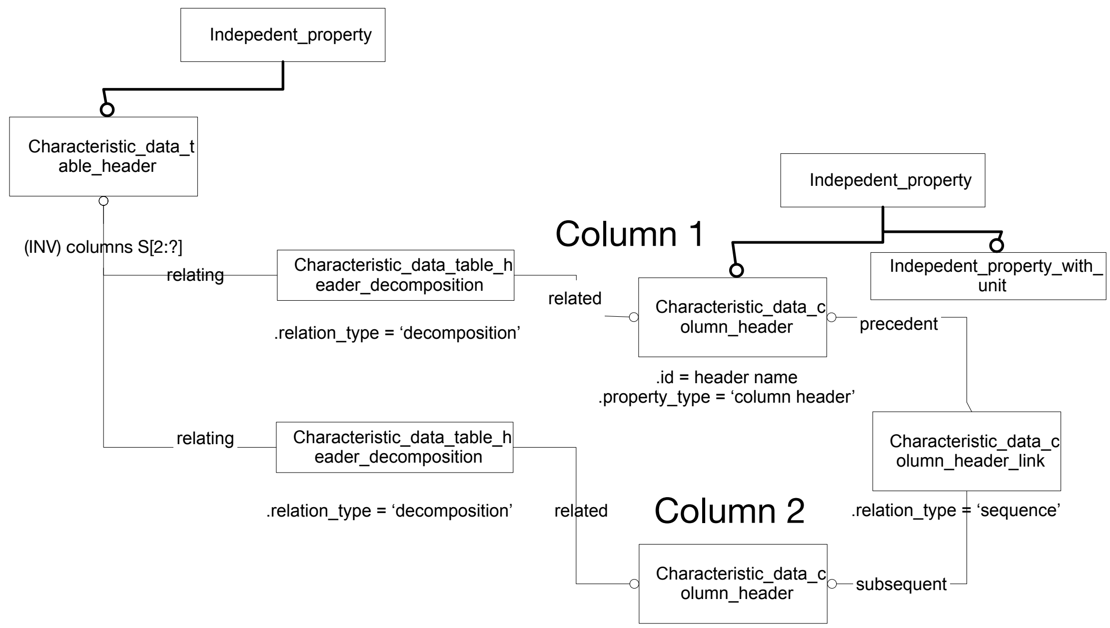

Figure 20. Characteristic_data_table_header AO type hierarchy

Figure 20 shows that individual column headers are linked with an explicit link element (Characteristic_data_column_header_link). Also illustrated is that Characteristic_data_table_header is a subtype of Independent_property as is Characteristic_data_column_header. The cells within a row are organized solely by list index position so that the column headers may be used in more than one table. It is recommended that the column header sequence be based on index position, with the implication that the column sequence and the index position data must be consistent. An approach is provided to associate a unit with a table column header. The recommendation at the ARM level is provided in the combined entity and instance diagram in Figure 21. Note that the combination block "Characteristic_data_column_header + Independent_property_with_unit instance" is a complex instance. Independent_property, Characteristic_data_table_header and Independent_property_with_unit highlight the inheritance structure. The corresponding recommended MIM instance diagram is provided in Figure 22. There are instantiation rules specified in the mapping specification for Independent_property_with_unit. In instance #25, representation.name = 'allowed units'. In instance #27, descriptive_representation_item.description = 'allowed units'.

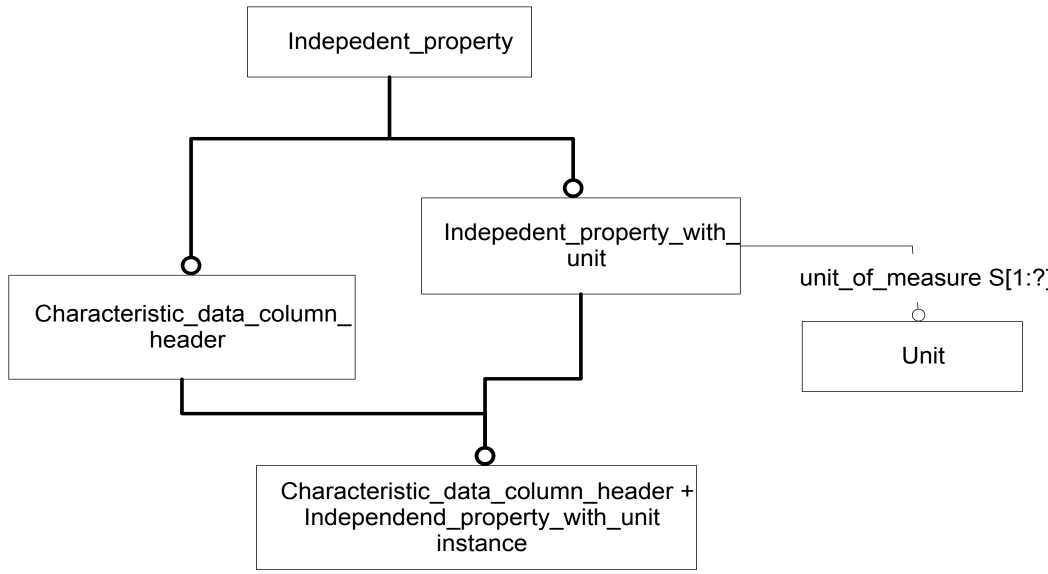

Figure 21. ARM type hierarchy for Characteristic_data_column_header

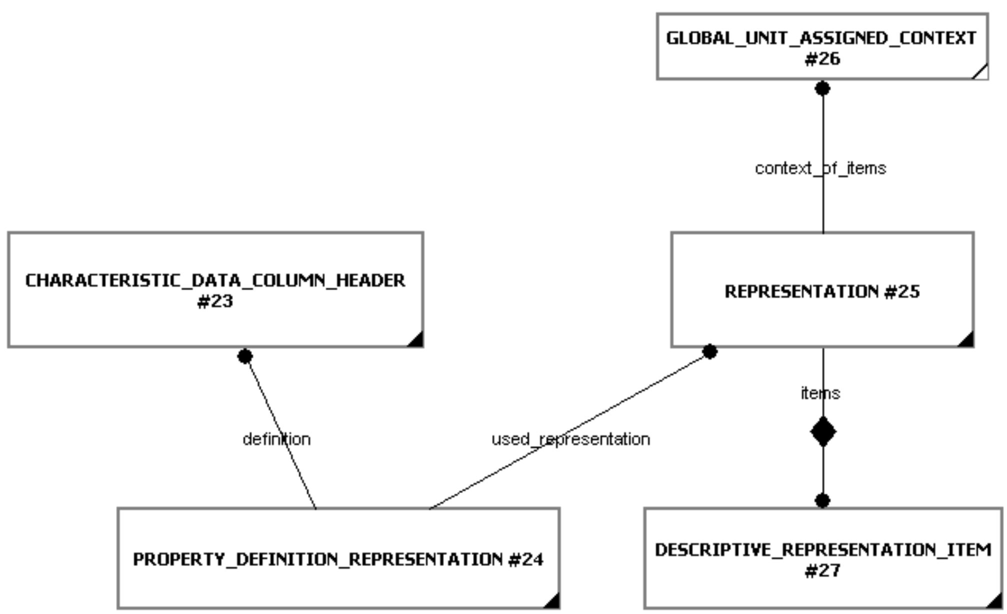

Figure 22. MIM instance diagram illustration for adding a unit to a column header

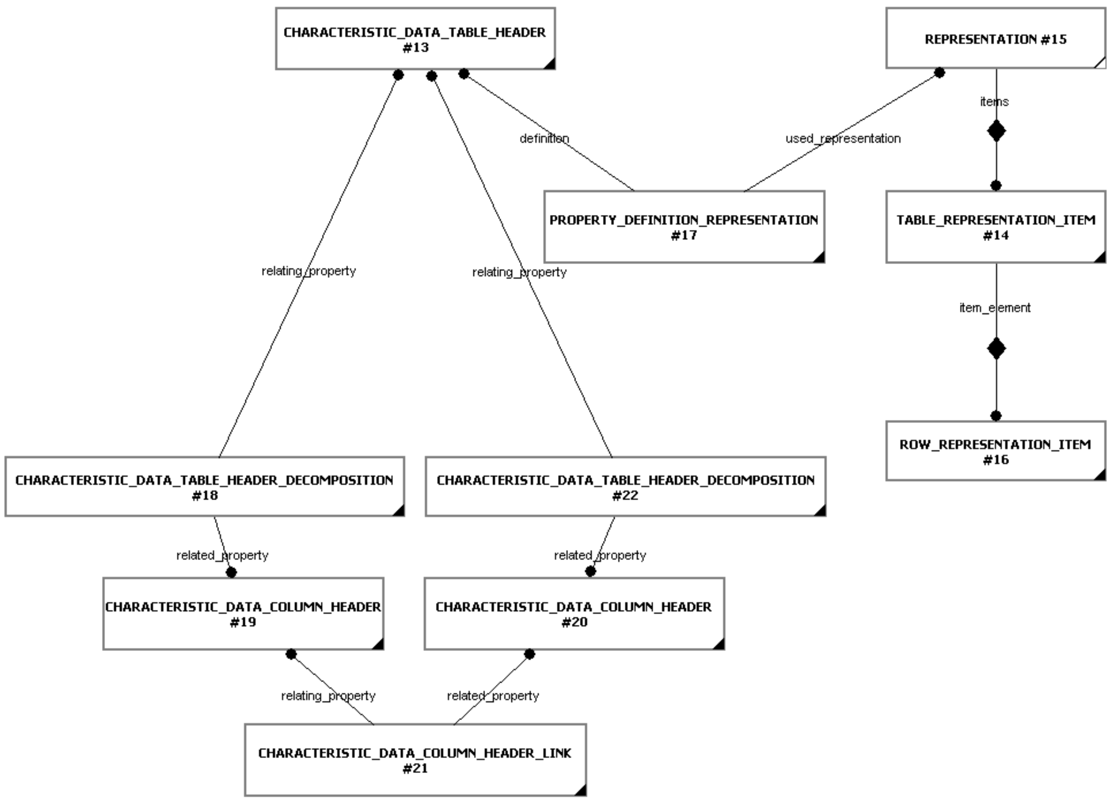

Figure 23. MIM instance diagram and mapping of Characteristic_data_table_header AO

In Figure 24, the possible value representations for a table cell are described.

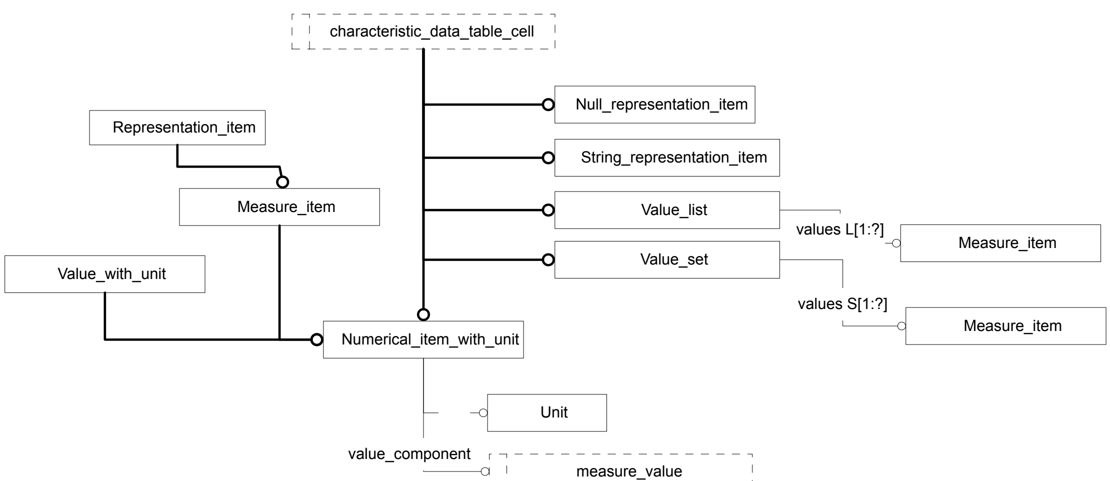

Figure 24. ARM type hierarchy for Numerical_item_with unit showing characteristic_data_cell options

The appendix contains a representative Part 21 physical file instantiation of the concepts discussed above. In this particular example, a number of characteristics have been populated for a National Semiconductor 250 MHz Low Noise Amplifier, part number LMH6654. There are 10 parameter_assignment entities in the file, all referencing a single product associated with the packaged_part. These parameter assignments are associated with 5 instances of model_parameter. The represented quantities are expressed in a variety of SI units, including V, °C, MHz, µA, and MΩ. The populated parameter assignments include the following:

Parameter_assignment	Description
#17	Supply Voltage Operating Rating expressed as a range characteristic
#39	Junction Temperature Operating Rating expressed as a range
#49, #63, #74, #85	Typical values of Closed-loop bandwidth at 4 different voltage conditions
#96	Typical and maximum values for input bias current at a condition of Vcm = 0V.
#111 and #128	Typical values for Input Resistance in Common Mode and Differential Mode operation (share same model parameter)
#139	A table representation of Input Bias Current as a function of Temperature (shares same model parameter as #96)

```
ISO-10303-21;
DATA;
#1=APPLICATION_CONTEXT('component model demonstration');
#2=APPLICATION_PROTOCOL_DEFINITION('INTERNATIONAL
STANDARD','ap210_electronic_assembly_interconnect_and_packaging_design',
     2014,#1);
#3=PACKAGED_PART('LMH6654MA',$,#5,#4,'',$,*);
#4=PRODUCT_DEFINITION_CONTEXT('part definition',#1,'');
#5=PRODUCT_DEFINITION_FORMATION('June 24, 2009',$,#6);
#6=PRODUCT('LMH6654MA','',$,(#8));
#7=PRODUCT_RELATED_PRODUCT_CATEGORY('part',$,(#6));
#8=PRODUCT_CONTEXT('part',#1,'');
#9=APPLIED_CLASSIFICATION_ASSIGNMENT(#11,#10,(#6));
#10=CLASSIFICATION_ROLE('','packaged part classification');
#11=CLASS('Low Noise Amplifier',$);
#12=APPLIED_ORGANIZATION_ASSIGNMENT(#13,#14,(#6));
#13=ORGANIZATION($,'National Semiconductor',$);
#14=ORGANIZATION_ROLE('owner');
#15=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#16=PROPERTY_DEFINITION('assigned parameter',$,#15);
#17=PARAMETER_ASSIGNMENT(#19,#17,'range characteristic',(#21,#22,#33),#20);
#18=PROPERTY_DEFINITION_REPRESENTATION(#16,#17);
#19=MODEL_PARAMETER('Vs','Supply Voltage, V+ - V-','Operating Ratings');
#20=REPRESENTATION_CONTEXT('range characteristic context','');
#21=DESCRIPTIVE_REPRESENTATION_ITEM('range type','closed');
#22=(ELECTRIC_POTENTIAL_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(ELECTRIC_POTENTIAL_MEASURE(2.5),#24)
     QUALIFIED_REPRESENTATION_ITEM((#23))
     REPRESENTATION_ITEM(''));
#23=TYPE_QUALIFIER('minimum');
#24=SI_ELECTRIC_POTENTIAL_UNIT((#25,#27,#29,#31),*,$,.VOLT.);
#25=DERIVED_UNIT_ELEMENT(#26,2.0);
#26=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT($,.METRE.));
#27=DERIVED_UNIT_ELEMENT(#28,1.0);
#28=(MASS_UNIT()NAMED_UNIT(*)SI_UNIT(.KILO.,.GRAM.));
#29=DERIVED_UNIT_ELEMENT(#30,-3.0);
#30=(NAMED_UNIT(*)SI_UNIT($,.SECOND.)TIME_UNIT());
#31=DERIVED_UNIT_ELEMENT(#32,-1.0);
#32=(ELECTRIC_CURRENT_UNIT()NAMED_UNIT(*)SI_UNIT($,.AMPERE.));
#320=(ELECTRIC_CURRENT_UNIT()NAMED_UNIT(*)SI_UNIT(.MICRO.,.AMPERE.));
#33=(ELECTRIC_POTENTIAL_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(ELECTRIC_POTENTIAL_MEASURE(6.0),#24)
     QUALIFIED_REPRESENTATION_ITEM((#34))
     REPRESENTATION_ITEM(''));
#34=TYPE_QUALIFIER('maximum');

#35=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#36=PROPERTY_DEFINITION('assigned parameter',$,#35);
#37=PARAMETER_ASSIGNMENT(#39,#37,'range characteristic',(#41,#42,#45),#40);
#38=PROPERTY_DEFINITION_REPRESENTATION(#36,#37);
#39=MODEL_PARAMETER('Tj','Junction Temperature Range','Operating Ratings');
#40=REPRESENTATION_CONTEXT('range characteristic context','');
#41=DESCRIPTIVE_REPRESENTATION_ITEM('range type','closed');
#42=(MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(THERMODYNAMIC_TEMPERATURE_MEASURE(85.0),#44)
     QUALIFIED_REPRESENTATION_ITEM((#43))
     REPRESENTATION_ITEM('')
     THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT());
#43=TYPE_QUALIFIER('maximum');
#44=(NAMED_UNIT(*)SI_UNIT($,.DEGREE_CELSIUS.)THERMODYNAMIC_TEMPERATURE_UNIT());
```

```
#45=(MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(THERMODYNAMIC_TEMPERATURE_MEASURE(-40.0),#44)
     QUALIFIED_REPRESENTATION_ITEM((#46))REPRESENTATION_ITEM('')
     THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT());
#46=TYPE_QUALIFIER('minimum');

#47=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#48=PROPERTY_DEFINITION('assigned parameter',$,#47);
#49=MATERIAL_PROPERTY_REPRESENTATION(#500)
    PARAMETER_ASSIGNMENT()
    PROPERTY_DEFINITION_REPRESENTATION(#51,#49)
    REPRESENTATION('tolerance',(#57),#56);
#500=DATA_ENVIRONMENT('parameter assignment conditions',$,(#17,#501));
#501=PROPERTY_DEFINITION_REPRESENTATION(#502,#53);
#502=GENERAL_PROPERTY('measurement conditions','measurement conditions',$);
#51=MODEL_PARAMETER('f_CL','Close Loop Bandwidth','+/-5V Electrical
Characteristics');
#53=REPRESENTATION('conditions',(#55),#54);
#54=REPRESENTATION_CONTEXT('conditions','');
#55=(ELECTRIC_POTENTIAL_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(ELECTRIC_POTENTIAL_MEASURE(1.0),#24)
     REPRESENTATION_ITEM('A_V'));
#56=REPRESENTATION_CONTEXT('tolerance values','tolerance');
#57=(FREQUENCY_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(FREQUENCY_MEASURE(250.E6),#59)
     QUALIFIED_REPRESENTATION_ITEM((#58))
     REPRESENTATION_ITEM(''));
#58=TYPE_QUALIFIER('typical');
#59=SI_FREQUENCY_UNIT((#60),*,$,.HERTZ.);
#60=DERIVED_UNIT_ELEMENT(#30,-1.0);

#61=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#62=PROPERTY_DEFINITION('assigned parameter',$,#61);
#63=MATERIAL_PROPERTY_REPRESENTATION(#500)
    PARAMETER_ASSIGNMENT()
    PROPERTY_DEFINITION_REPRESENTATION(#62,#63);
    REPRESENTATION('tolerance',(#70),#69);
#68=(ELECTRIC_POTENTIAL_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(ELECTRIC_POTENTIAL_MEASURE(2.0),#24)
     REPRESENTATION_ITEM('A_V'));
#69=REPRESENTATION_CONTEXT('tolerance values','tolerance');
#70=(FREQUENCY_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(FREQUENCY_MEASURE(139.E6),#59)
     QUALIFIED_REPRESENTATION_ITEM((#71))
     REPRESENTATION_ITEM(''));
#71=TYPE_QUALIFIER('typical');

#72=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#73=PROPERTY_DEFINITION('assigned parameter',$,#72);

#74=MATERIAL_PROPERTY_REPRESENTATION(#500)
    PARAMETER_ASSIGNMENT();
    PROPERTY_DEFINITION_REPRESENTATION(#51,#74);
    REPRESENTATION('tolerance',(#81),#80);
#79=(ELECTRIC_POTENTIAL_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(ELECTRIC_POTENTIAL_MEASURE(5.0),#24)
     REPRESENTATION_ITEM('A_V'));
#80=REPRESENTATION_CONTEXT('tolerance values','tolerance');
```

```
#81=(FREQUENCY_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(FREQUENCY_MEASURE(52.E6),#59)
     QUALIFIED_REPRESENTATION_ITEM((#82))
     REPRESENTATION_ITEM(''));
#82=TYPE_QUALIFIER('typical');

#83=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#84=PROPERTY_DEFINITION('assigned parameter',$,#83);
#85=MATERIAL_PROPERTY_REPRESENTATION(#500)
    PARAMETER_ASSIGNMENT();
    PROPERTY_DEFINITION_REPRESENTATION(#51,#85);
    REPRESENTATION('tolerance',(#92),#91);
#90=(ELECTRIC_POTENTIAL_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(ELECTRIC_POTENTIAL_MEASURE(10.0),#24)
     REPRESENTATION_ITEM('A_V'));
#91=REPRESENTATION_CONTEXT('tolerance values','tolerance');
#92=(FREQUENCY_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(FREQUENCY_MEASURE(26.E6),#59)
     QUALIFIED_REPRESENTATION_ITEM((#93))
     REPRESENTATION_ITEM(''));
#93=TYPE_QUALIFIER('typical');

#94=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#95=PROPERTY_DEFINITION('assigned parameter',$,#94);
#96=MATERIAL_PROPERTY_REPRESENTATION(#500)
    PARAMETER_ASSIGNMENT();
    PROPERTY_DEFINITION_REPRESENTATION(#98,#96);
    REPRESENTATION('tolerance',(#105,#107),#104);
#98=MODEL_PARAMETER('Ib','Input Bias Current','');
#102=DESCRIPTIVE_REPRESENTATION_ITEM('','+/-5V Electrical Characteristics');
#103=(ELECTRIC_POTENTIAL_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(ELECTRIC_POTENTIAL_MEASURE(0.0),#24)
      REPRESENTATION_ITEM('Vcm'));
#104=REPRESENTATION_CONTEXT('tolerance values','tolerance');
#105=(ELECTRIC_CURRENT_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(ELECTRIC_CURRENT_MEASURE(12.0),#320)
      QUALIFIED_REPRESENTATION_ITEM((#106))
      REPRESENTATION_ITEM(''));
#106=TYPE_QUALIFIER('maximum');
#107=(ELECTRIC_CURRENT_MEASURE_WITH_UNIT(
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(ELECTRIC_CURRENT_MEASURE(5.0),#320)
      QUALIFIED_REPRESENTATION_ITEM((#108))
      REPRESENTATION_ITEM(''));
#108=TYPE_QUALIFIER('typical');

#109=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#110=PROPERTY_DEFINITION('assigned parameter',$,#109);
#113=MODEL_PARAMETER('Rin','Input Resistance','+/-5V Electrical
Characteristics');
#111=MATERIAL_PROPERTY_REPRESENTATION(#1110)
     PARAMETER_ASSIGNMENT();
     PROPERTY_DEFINITION_REPRESENTATION(#113,#111);
     REPRESENTATION('tolerance',(#119),#1110);
#1110=DATA_ENVIRONMENT('parameter assignment conditions',$,(#17,#112));
#112=PROPERTY_DEFINITION_REPRESENTATION(#117,#115);
#117=GENERAL_PROPERTY('Mode','Common','conditions');
#115=REPRESENTATION('Common mode measurement condition values',(#1170),#116);
#116=REPRESENTATION_CONTEXT('conditions','');
```

```
#1170=DESCRIPTIVE_REPRESENTATION_ITEM('common mode condition values not
shown');
#119=(MEASURE_REPRESENTATION_ITEM(
      MEASURE_WITH_UNIT(RESISTANCE_MEASURE(4.E6),#121)
      QUALIFIED_REPRESENTATION_ITEM((#120))
      REPRESENTATION_ITEM('')
      RESISTANCE_MEASURE_WITH_UNIT());
#120=TYPE_QUALIFIER('typical');
#121=SI_RESISTANCE_UNIT((#122,#123,#124,#125),*,$,.OHM.);
#122=DERIVED_UNIT_ELEMENT(#26,2.0);
#123=DERIVED_UNIT_ELEMENT(#28,1.0);
#124=DERIVED_UNIT_ELEMENT(#30,-3.0);
#125=DERIVED_UNIT_ELEMENT(#32,-2.0);

#126=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#127=PROPERTY_DEFINITION('assigned parameter',$,#126);
#128=MATERIAL_PROPERTY_REPRESENTATION(#1280)
     PARAMETER_ASSIGNMENT();
     PROPERTY_DEFINITION_REPRESENTATION(#113,#128);
     REPRESENTATION('tolerance',(#135),#1280);
#1280=DATA_ENVIRONMENT('differential mode parameter assignment conditions',$,
(#17,#129));
#129=PROPERTY_DEFINITION_REPRESENTATION(#1290,#131);
#1290=GENERAL_PROPERTY('differential mode parameter assignment
conditions','Differential',$);
#131=REPRESENTATION('differential mode parameter assignment conditions',
(#133),#132);
#132=REPRESENTATION_CONTEXT('conditions','');
#133=DESCRIPTIVE_REPRESENTATION_ITEM('Mode','Differential conditions values');
#135=(MEASURE_REPRESENTATION_ITEM(
     MEASURE_WITH_UNIT(RESISTANCE_MEASURE(4.E6),#121)
     QUALIFIED_REPRESENTATION_ITEM((#136))REPRESENTATION_ITEM('')
     RESISTANCE_MEASURE_WITH_UNIT());
#136=TYPE_QUALIFIER('typical');

#137=PRODUCT_SPECIFIC_PARAMETER_VALUE_ASSIGNMENT('',$,'',$,(#6));
#138=PROPERTY_DEFINITION('assigned parameter',$,#137);
#139=MATERIAL_PROPERTY_REPRESENTATION(#1390)
     PARAMETER_ASSIGNMENT();
     PROPERTY_DEFINITION_REPRESENTATION(#98,#139);
     REPRESENTATION('characteristic data table',(#154),#145);
#1390=DATA_ENVIRONMENT('parameter assignment conditions for data table',$,
(#17,#140));
#140=PROPERTY_DEFINITION_REPRESENTATION(#1400,#142);
#1400=GENERAL_PROPERTY('conditions for data table','conditions for data table',
$);
#142=REPRESENTATION('conditions for data table',(#144),#143);
#143=REPRESENTATION_CONTEXT('conditions for data table','');
#144=DESCRIPTIVE_REPRESENTATION_ITEM('','Typical Performance Characteristics');
#145=REPRESENTATION_CONTEXT('numerical representation context','');

#146=CHARACTERISTIC_DATA_TABLE_HEADER('','',$);
#147=PROPERTY_DEFINITION_REPRESENTATION(#146,#139);
#148=NAME_ATTRIBUTE('table header',#147);

#149=CHARACTERISTIC_DATA_COLUMN_HEADER('Temperature','column header',$);
#150=CHARACTERISTIC_DATA_TABLE_HEADER_DECOMPOSITION('decomposition',
$,#146,#149);
#151=CHARACTERISTIC_DATA_COLUMN_HEADER('Bias Current','column header',$);
#152=CHARACTERISTIC_DATA_TABLE_HEADER_DECOMPOSITION('decomposition',
$,#146,#151);
#153=CHARACTERISTIC_DATA_COLUMN_HEADER_LINK('sequence',$,#149,#151);

#154=TABLE_REPRESENTATION_ITEM
```

```
     ('table',LIST_REPRESENTATION_ITEM((#155,#158,#161,#164)));

#155=ROW_REPRESENTATION_ITEM('row1',LIST_REPRESENTATION_ITEM((#156,#157)));
#156=(MEASURE_REPRESENTATION_ITEM(
     MEASURE_WITH_UNIT(THERMODYNAMIC_TEMPERATURE_MEASURE(-40.0),#44)
     REPRESENTATION_ITEM('')
     THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT());
#157=(ELECTRIC_CURRENT_MEASURE_WITH_UNIT()
     MEASURE_REPRESENTATION_ITEM()
     MEASURE_WITH_UNIT(ELECTRIC_CURRENT_MEASURE(6.7),#320
     REPRESENTATION_ITEM(''));

#158=ROW_REPRESENTATION_ITEM('row2',LIST_REPRESENTATION_ITEM((#159,#160)));
#159=(MEASURE_REPRESENTATION_ITEM(
      MEASURE_WITH_UNIT(THERMODYNAMIC_TEMPERATURE_MEASURE(0.0),#44
      REPRESENTATION_ITEM('')
      THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT());
#160=(ELECTRIC_CURRENT_MEASURE_WITH_UNIT(
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(ELECTRIC_CURRENT_MEASURE(6.2),#320
      REPRESENTATION_ITEM(''));

#161=ROW_REPRESENTATION_ITEM('row3',LIST_REPRESENTATION_ITEM((#162,#163)));
#162=(MEASURE_REPRESENTATION_ITEM(
      MEASURE_WITH_UNIT(THERMODYNAMIC_TEMPERATURE_MEASURE(50.0),#44
      REPRESENTATION_ITEM('')
      THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT());
#163=(ELECTRIC_CURRENT_MEASURE_WITH_UNIT(
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(ELECTRIC_CURRENT_MEASURE(5.5),#320
      REPRESENTATION_ITEM(''));

#164=ROW_REPRESENTATION_ITEM('row4',LIST_REPRESENTATION_ITEM((#165,#166)));
#165=(MEASURE_REPRESENTATION_ITEM(
      MEASURE_WITH_UNIT(THERMODYNAMIC_TEMPERATURE_MEASURE(75.0),#44
      REPRESENTATION_ITEM('')
      THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT());
#166=(ELECTRIC_CURRENT_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(ELECTRIC_CURRENT_MEASURE(5.0),#320
      REPRESENTATION_ITEM(''));
ENDSEC;
END-ISO-10303-21;
```