AP214 3rd Edition

Summary of the GD&T Changes for AP214

PROSTEP AG
Doliviostraße 11
64293 Darmstadt
Tel.: +49 (0) 6151 / 9287-0
Fax: +49 (0) 6151 / 9287 - 326

PROSTEP AG
Dolivostraße 11
64293 Darmstadt
Tel. +49-61 51-92 87-0
Fax +49-61 51-92 87-3 26

www.prostep.com

Geschäftsstelle Berlin
PROSTEP GIDA GmbH
Albert-Einstein-Straße 16
12489 Berlin
Tel. +49-30-63 92-60 30
Fax +49-30-63 92-60 50

Geschäftsstelle Hannover
Karl-Wiechert-Allee 72
30625 Hannover
Tel. +49-5 11-5 40 58-0
Fax +49-5 11-5 40 58-1 50

Geschäftsstelle München
Taunusstraße 42
80807 München
Tel. +49-89-3 50 20-0
Fax +49-89-3 50 20-2 00

Geschäftsstelle Stuttgart
Curiestraße 2
70563 Stuttgart
Tel. +49-7 11-67 40 03 48
Fax +49-7 11-67 40 02 00

Bankverbindung
Bayerische Hypo- und Vereinsbank
Konto-Nr. 300 51 00
BLZ 508 202 92

Vorstand
Dr. Bernd Pätzold (Vorsitz)
Reinhard Betz

Aufsichtsratsvorsitz
Dr. Heinz-Gerd Lehnhoff

Handelsregister
Amtsgericht Darmstadt
HRB 8383
Ust.-ID-Nr: DE 164374342

Contents

1. General modifications concerning various clauses

1.1 Change all occurrences of the term “compound datum” by “common datum”

- 4.1.31 geometric_tolerance (T2)
- Figure 20 – Usage of compound datums
- 5.2.1.11 Dimensioning and Tolerances
- Figure 83 – Use of common_datum

Check for all references for the renamed AO Compound_datum (see 2.1.8 below)

2. Clause 4.2 Application Object Definition

2.1 Modified AO definitions

2.1.1 General_size_dimension

- Rename to Externally_defined_size_dimension
- Rename attribute dimension_type to name
- require document assignment for externally_defined_size_dimension, same way as for e.g. Catalogue_thread

4.2.232 Externally_defined_size_dimension

An Externally_defined_size_dimension is the definition of a dimension that is defined in an external document. Each Externally_defined_size_dimension shall be referenced by a Document_assignment.

An Externally_defined_size_dimension is a type of Size_dimension.

The data associated with an Externally_defined_size_dimension are the following:

- name;
- used_path.

4.2.232.1 name

The name specifies the word or group of words used to refer to the Externally_defined_size_dimension.

4.2.232.2 used_path

The used_path specifies the Measurement_path along which the size is applied or measured.

See Externally_defined_size_dimension to Measurement_path for the application assertion.

The used_path need not be specified for a particular Externally_defined_size_dimension .

2.1.2 Value_limitation

- Move attribute envelope to Size_dimension
- Revise definition to be in compliance with AM1050

4.2.545.1 envelope

The envelope specifies whether the envelope principle applies or not. A value of 'true' indicates that the envelope principle applies to this size dimension. When the envelope principle applies, the envelope of the perfect shape corresponding to the maximum material shall not be larger than the specified dimension and tolerance. The envelope principle specifies that size and form tolerances are not additive.

NOTE 1 - The concept of the envelope principle is defined in ISO 1101 [XX].

NOTE 2 - A dimension where the envelope principle applies, appears on drawings with an 'E' enclosed in a circle following the dimension value.

NOTE 3 The envelope principle may have counterparts in other standards.

EXAMPLE The envelope principle is equivalent to ASME Y14.5 rule 1

The envelope need not be specified for a particular Value_limitation.

2.1.3 *Size_dimension*

- Add NOTE in 4.2.456 Size_dimension:

NOTE Size_dimension as defined in this part of ISO 10303 describe an intrinsic length or angle characteristic of part shape. This includes characteristics of what are traditionally called “features-of-size”, but they are not limited to that usage alone. For example, the radial_size_dimension and curved_size_dimension application objects describe characteristics which may not be applied to “features-of-size”.

2.1.4 *Linear_distance_dimension*

- Move attribute “directed” to Supertype Location_dimension

2.1.5 *Angular_location_dimension*

- Delete attributes major_angle, mirror_target, origin_termination
- Delete Figure 9, NOTE and EXAMPLE
- Add new attribute orientation

4.2.13 Angular_location_dimension

An Angular_location_dimension is the representation of a geometrical angular constraint concerning the position of two elements.

An Angular_location_dimension is a type of Location_dimension.

The data associated with an Angular_location_dimension are the following:

- orientation.

4.2.13.1 orientation

The orientation specifies the direction of positive measure of the Angular_location_dimension. The angle extends in the XY plane from the X axis towards Y, about the origin.

2.1.6 *Geometric_dimension*

- Add new attribute dimension_note

4.2.237 Geometric_dimension

A Geometric_dimension is the definition of the size of an element or of the distance between two elements.

NOTE A Geometric_dimension may be presented by a Dimension_callout showing the dimension value and its relationship to the drafted geometry or annotation.

Each Geometric_dimension is a Size_dimension or a Location_dimension.

The data associated with a Geometric_dimension are the following:

- notes;
- dimension_value;
- id.

4.2.237.1 notes

The notes specifies qualifying notse that provide additional information about the Geometric_dimension.

Where applicable the following values shall be used:

- "auxiliary": the dimension is for information purposes only. The dimension_value shall be a Numerical_value;

§ "theoretical": the dimension is theoretically exact. The dimension_value shall be a Numerical_value .

There may be more than zero notes for a Geometric_dimension.

4.2.237.2 dimension_value

The dimension_value specifies a positive value that is the value of size of the element or of the distance between the two elements.

See Geometric_dimension to Value_with_unit for the application assertion.

4.2.237.3 id

The id specifies the identifier of the Geometric_dimension.

2.1.7 Document_assignment

¬ Attribute role: Add “dimensioning standard” to the permissive list values

4.2.156.3 role

The role specifies the meaning of the Document_assignment.

Where applicable the following values shall be used:

§ 'additional information': The assigned document provides information that is relevant for the associated object, but is not a description of the associated object itself;

§ 'behavior': The assigned document specifies information about the behaviour of the associated object;

§ 'description': The assigned document provides textual information for the associated object itself;

§ 'dimensioning standard': The assigned document specifies the dimensioning standard;

NOTE The document shall be assigned to a Design_discipline_item_definition.

§ 'informative': The assigned document may or may not be considered;

§ 'mandatory': The associated object shall conform to the content of the assigned document;

§ 'mathematical description': The assigned document specifies the associated object by providing the algorithmic specification of its behavior.

EXAMPLE For a Product_component there is a textual document assigned, explaining the functionality, size, and other characteristics by a Document_assignment with role 'description'. Another document, representing the specification of the Product_class, in which the Product_component is used, is also assigned to the same Product_component by a Document_assignment with role 'additional information'.

2.1.8 Compound_datum

¬ Rename to Common_datum; replace all occurrences of Compound_datum to the new term:

- ¬ Clause 4.2, 4.3, 5.1, Annexes, definitions (clause 3)

2.1.9 Datum

¬ Move attribute precedence to new AO Datum_reference

2.1.10 Single_datum

¬ Remove attribute is_defined_by

2.1.11 References to Datum

¬ Change all references to Datum to Datum_reference:

- ¬ perpendicularity_tolerance
- ¬ parallelism_tolerance
- ¬ angularity_tolerance
- ¬ concentricity_tolerance
- ¬ symmetry_tolerance
- ¬ position_tolerance

- circular_runout_tolerance
- total_runout_tolerance
- line_profile_tolerance
- surface_profile_tolerance
- coaxiality_tolerance
- Replace the definition of the attribute reference_datum by the following text:

4.2.345.2 reference_datum

The reference_datum specifies the Datum_reference objects that define the reference for this type of geometric tolerance.

2.2 New Application Objects

2.2.1 Curved_size_dimension

4.2.x Curved_size_dimension

A Curved_size_dimension is the definition of a dimension that is the tolerance on a dimension for a curve measured along the entire path of a curve.

A Curved_size_dimension is a type of Size_dimension.

2.2.2 Diameter_size_dimension

4.2.x Diameter_size_dimension

A Diameter_size_dimension is the definition of a dimension that is a diameter of a circular, cylindrical, or spherical feature.

A Diameter_size_dimension is a type of Size_dimension.

2.2.3 Height_size_dimension

4.2.x Height_size_dimension

A Hight_size_dimension is the definition of a dimension that may be used to represent the height of a feature.

A Hight_size_dimension is a type of Size_dimension.

The data associated with a Hight_size_dimension are the following:

- used_path.

4.2.x.1 used_path

The used_path specifies the Measurement_path along which the size is applied or measured.

See Hight_size_dimension to Measurement_path for the application assertion.

The used_path need not be specified for a particular Hight_size_dimension .

2.2.4 Length_size_dimension

4.2.x Length_size_dimension

A Length_size_dimension is the definition of a dimension that may be used to represent the length of a feature.

A Length_size_dimension is a type of Size_dimension.

The data associated with a Length _size_dimension are the following:

- used_path.

4.2.x.1 used_path

The used_path specifies the Measurement_path along which the size is applied or measured.

See Length_size_dimension to Measurement_path for the application assertion.

The used_path need not be specified for a particular Length_size_dimension .

2.2.5 Radial_size_dimension

- + additional optional attribute radius_type

4.2.x Radial_size_dimension

A Radial_size_dimension is the definition of a dimension that is a radius of a circular, cylindrical, or spherical feature.

A Radial_size_dimension is a type of Size_dimension.

The data associated with a Radial_size_dimension are the following:

- radius_type.

4.2.x.1 radius_type

The radius_type specifies the form of the tolerance zone.

The following values shall be used:

- 'adjoining: The tolerance zone is a crescent moon shape bounded by a minimum radius arc and a maximum radius arc, in which each arc shall be tangent to the adjacent surfaces.radius is located by its end points;

NOTE 1 An adjoining radial tolerance typically applied to radial surfaces, such as filleted surfaces, that are located by adjoining surfaces. It is drawn without a center and the radial leader just points to the radial surface.

- "centred": The tolerance zone is the shape in between two annular radial segments with their centers at the prescribed center point. The radial tolerance permits steps in the tolerance zones with respect to its adjoining surface.

NOTE 2 A centered radial tolerance is typically applied to radial surfaces, such as filleted surfaces, that are located by center point. It is drawn with a center and the radial leader passes through the center.

The value of this attribute need not be specified. If omitted, a value of 'adjoining' shall be assumed.

2.2.6 Width_size_dimension

4.2.x Width_size_dimension

A Width_size_dimension is the definition of a dimension that may be used to represent the width of a feature.

A Width_size_dimension is a type of Size_dimension.

The data associated with a Width_size_dimension are the following:

- used_path.

4.2.x.1 used_path

The used_path specifies the Measurement_path along which the size is applied or measured.

See Width_size_dimension to Measurement_path for the application assertion.

The used_path need not be specified for a particular Width_size_dimension .

2.2.7 Datum_reference

4.2.x Datum_reference

A Datum_reference is the usage of a Datum by a geometric tolerance.

The data associated with a Datum are the following:

- Precedence;
- referenced_datum.

4.2.x.1 precedence

The precedence specifies the priority assigned to this instance of Datum_reference when it is referenced by a geometric tolerance.

4.2.x.2 referenced_datum

The referenced_datum specifies the instance of Datum that is referenced by the Datum_reference.

See Datum_reference to Datum for the application assertion.

2.2.8 Datum_defined_by_derived_shape

4.2.x Datum_defined_by_derived_shape

A Datum_defined_by_derived_shape is a datum that is represented by a Derived_geometry.

A Datum_defined_by_derived_shape is a type of Single_datum and a type of Derived_geometry.

2.2.9 Datum_defined_by_feature

4.2.x Datum_defined_by_feature

A Datum_defined_by_feature is a datum that is represented by a Shape_element.

A Datum_defined_by_feature is a type of Single_datum.

The data associated with a Datum_defined_by_feature are the following:

- defined_by.

4.2.x.1 defined_by

The defined_by specifies the Shape_element that represents the Datum_defined_by_feature..

2.2.10 Datum_defined_by_targets

4.2.x Datum_defined_by_targets

A Datum_defined_by_targets is a datum that is represented by a set of Datum_target.

A Datum_defined_by_targets is a type of Single_datum.

NOTE Datums are established from a set of datum targets when the use of an entire feature would introduce excessive variations or lack of repeatability in measurements.

The data associated with a Datum_defined_by_targets are the following:

- defined_by;
- rule_description.

4.2.x.1 defined_by

The defined_by specifies the set of Datum_target that represents the Datum_defined_by_targets.

4.2.x.2 rule_description

The rule_description specifies the type of datum that is formed by the Datum_defined_by_targets.

EXAMPLE The rule_description 'V-block' indicates that two Datum_target objects on a cylindrical element are supposed to form the areas of contact in a V-shaped fixture.

See Datum_defined_by_target_set to Multi_language_string for the application assertion.

The rule_description need not be specified for a particular Datum_defined_by_target_set.

If present, there shall be exactly one object that defines the rule_description for a Datum_defined_by_target_set.

2.3 Deleted AO

2.3.1 Datum_target_set

3. Clause 4.3 Application Assertions

3.1 New Assertions

3.1.1 Document_assignment to Externally_defined_size_dimension

- Add Externally_defined_size_dimension to Documented_element_select
- Add new asertion

4.3.xxx Document_assignment to Externally_defined_size_dimension

Each Document_assignment is_assigned_to zero or one Externally_defined_size_dimension. Each Externally_defined_size_dimension is related to zero or more Document_assignment objects.

NOTE - This assertion is established through documented_element_select.

4. Clause 5.1 Mapping Table

4.1 Modified Mappings

4.1.1 Value_range

- Add OR case mapping
- Propagate changed mapping to value_with_unit and property_value

VALUE_RANGE #1: If no significant digits are given for the value_with_unit.:: #2: If significant digits are given for the value_with_unit.::: #3: If the unit is not assigned globally.:: #4: If the unit is assigned globally.:: #5: If referenced by a Geometric_dimension.:: #6: If not referenced by a Geometric_dimension.::	#6(#1(value_range) #2([value_range] [qualified_representation_item])) #5(shape_dimension_representation)	214 214 45 47		#6({value_range <= compound_representation_item})
lower_limit	#6(#3(measure_with_unit.value_component) #4(value_representation_item.value_component)) #5(PATH)	41 43	30	#6(value_range <= compound_representation_item compound_representation_item.item_element ->> set_representation_item #3(set_representation_item[i] ->> representation_item => {representation_item.name = 'lower limit'} measure_representation_item measure_representation_item <= measure_with_unit {(|measure_with_unit|) (measure_with_unit => (amount_of_substance_measure_with_unit) (area_measure_with_unit)

				(electric_current_measure_with_unit) (length_measure_with_unit) (luminous_intensity_measure_with_unit) (mass_measure_with_unit) (plane_angle_measure_with_unit) (ratio_measure_with_unit) (solid_angle_measure_with_unit) (thermodynamic_temperature_measure_with_unit) (time_measure_with_unit) (volume_measure_with_unit))} measure_with_unit.value_component {measure_with_unit.value_component -> measure_value (measure_value = area_measure area_measure) (measure_value = amount_of_substance_measure amount_of_substance_measure) (measure_value = length_measure length_measure) (measure_value = electric_current_measure electric_current_measure) (measure_value = plane_angle_measure plane_angle_measure) (measure_value = ratio_measure ratio_measure) (measure_value = parameter_value parameter_value) (measure_value = descriptive_measure descriptive_measure) (measure_value = positive_plane_angle_measure positive_plane_angle_measure) (measure_value = count_measure count_measure) (measure_value = mass_measure mass_measure) (measure_value = time_measure

				time_measure) (measure_value = thermodynamic_temperature_measure thermodynamic_temperature_measure) (measure_value = luminous_intensity_measure luminous_intensity_measure) (measure_value = solid_angle_measure solid_angle_measure) (measure_value = volume_measure volume_measure) (measure_value = numeric_measure numeric_measure) (measure_value = context_dependent_measure context_dependent_measure) (measure_value = positive_length_measure positive_length_measure) (measure_value = positive_ratio_measure positive_ratio_measure)}) #4(set_representation_item[i] ->) representation_item => {representation_item.name = 'lower limit'} value_representation_item value_representation_item.value_component)) #5(shape_dimension_representation <= shape_representation <= representation #3(representation.items[i] ->) {representation_item representation_item.name = 'lower range'} representation_item => measure_representation_item measure_representation_item <= measure_with_unit {(|measure_with_unit|) (measure_with_unit => (amount_of_substance_measure_with_unit) (area_measure_with_unit)

				(electric_current_measure_with_unit) (length_measure_with_unit) (luminous_intensity_measure_with_unit) (mass_measure_with_unit) (plane_angle_measure_with_unit) (ratio_measure_with_unit) (solid_angle_measure_with_unit) (thermodynamic_temperature_measure_with_unit) (time_measure_with_unit) (volume_measure_with_unit))} measure_with_unit.value_component {measure_with_unit.value_component -> measure_value (measure_value = area_measure area_measure) (measure_value = amount_of_substance_measure amount_of_substance_measure) (measure_value = length_measure length_measure) (measure_value = electric_current_measure electric_current_measure) (measure_value = plane_angle_measure plane_angle_measure) (measure_value = ratio_measure ratio_measure) (measure_value = parameter_value parameter_value) (measure_value = descriptive_measure descriptive_measure) (measure_value = positive_plane_angle_measure positive_plane_angle_measure) (measure_value = count_measure count_measure) (measure_value = mass_measure mass_measure) (measure_value = time_measure

				time_measure) (measure_value = thermodynamic_temperature_measure thermodynamic_temperature_measure) (measure_value = luminous_intensity_measure luminous_intensity_measure) (measure_value = solid_angle_measure solid_angle_measure) (measure_value = volume_measure volume_measure) (measure_value = numeric_measure numeric_measure) (measure_value = context_dependent_measure context_dependent_measure) (measure_value = positive_length_measure positive_length_measure) (measure_value = positive_ratio_measure positive_ratio_measure)})) #4(representation.items[i] ->) {representation_item representation_item.name = 'lower range'} representation_item => value_representation_item value_representation_item.value_component))
upper_limit	#6(#3(measure_with_unit.value_component) #4(value_representation_item.value_component)) #5(PATH)	41 43	30	#6(value_range <= compound_representation_item compound_representation_item.item_element ->) set_representation_item #3(set_representation_item[i] ->) representation_item => {representation_item.name = 'lower limit'} measure_representation_item measure_representation_item <= measure_with_unit {(|measure_with_unit|) (measure_with_unit => (amount_of_substance_measure_with_unit)

				(area_measure_with_unit) (electric_current_measure_with_unit) (length_measure_with_unit) (luminous_intensity_measure_with_unit) (mass_measure_with_unit) (plane_angle_measure_with_unit) (ratio_measure_with_unit) (solid_angle_measure_with_unit) (thermodynamic_temperature_measure_with_unit) (time_measure_with_unit) (volume_measure_with_unit))} measure_with_unit.value_component {measure_with_unit.value_component -> measure_value (measure_value = area_measure area_measure) (measure_value = amount_of_substance_measure amount_of_substance_measure) (measure_value = length_measure length_measure) (measure_value = electric_current_measure electric_current_measure) (measure_value = plane_angle_measure plane_angle_measure) (measure_value = ratio_measure ratio_measure) (measure_value = parameter_value parameter_value) (measure_value = descriptive_measure descriptive_measure) (measure_value = positive_plane_angle_measure positive_plane_angle_measure) (measure_value = count_measure count_measure) (measure_value = mass_measure mass_measure)

				(measure_value = time_measure time_measure) (measure_value = thermodynamic_temperature_measure thermodynamic_temperature_measure) (measure_value = luminous_intensity_measure luminous_intensity_measure) (measure_value = solid_angle_measure solid_angle_measure) (measure_value = volume_measure volume_measure) (measure_value = numeric_measure numeric_measure) (measure_value = context_dependent_measure context_dependent_measure) (measure_value = positive_length_measure positive_length_measure) (measure_value = positive_ratio_measure positive_ratio_measure)}) #4(set_representation_item[i] ->) representation_item => {representation_item.name = 'lower limit'} value_representation_item value_representation_item.value_component)) #5(shape_dimension_representation <= shape_representation <= representation #3(representation.items[i] ->) {representation_item representation_item.name = 'upper range'} representation_item => measure_representation_item measure_representation_item <= measure_with_unit {(|measure_with_unit|) (measure_with_unit => (amount_of_substance_measure_with_unit)

				(area_measure_with_unit) (electric_current_measure_with_unit) (length_measure_with_unit) (luminous_intensity_measure_with_unit) (mass_measure_with_unit) (plane_angle_measure_with_unit) (ratio_measure_with_unit) (solid_angle_measure_with_unit) (thermodynamic_temperature_measure_with_unit) (time_measure_with_unit) (volume_measure_with_unit))} measure_with_unit.value_component {measure_with_unit.value_component -> measure_value (measure_value = area_measure area_measure) (measure_value = amount_of_substance_measure amount_of_substance_measure) (measure_value = length_measure length_measure) (measure_value = electric_current_measure electric_current_measure) (measure_value = plane_angle_measure plane_angle_measure) (measure_value = ratio_measure ratio_measure) (measure_value = parameter_value parameter_value) (measure_value = descriptive_measure descriptive_measure) (measure_value = positive_plane_angle_measure positive_plane_angle_measure) (measure_value = count_measure count_measure) (measure_value = mass_measure mass_measure)

				(measure_value = time_measure time_measure) (measure_value = thermodynamic_temperature_measure thermodynamic_temperature_measure) (measure_value = luminous_intensity_measure luminous_intensity_measure) (measure_value = solid_angle_measure solid_angle_measure) (measure_value = volume_measure volume_measure) (measure_value = numeric_measure numeric_measure) (measure_value = context_dependent_measure context_dependent_measure) (measure_value = positive_length_measure positive_length_measure) (measure_value = positive_ratio_measure positive_ratio_measure)})) #4(representation.items[i] ->{representation_item representation_item.name = 'upper range'} representation_item => value_representation_item value_representation_item.value_component))

VALUE_WITH_UNIT #1: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.:: #2: If the value_with_unit is a value_range.:: #3: If no significant digits are given for the value_with_unit.:: #4: If significant digits are given for the value_with_unit.::	#1(#3(measure_representation_item) #4([measure_representation_item] [qualified_representation_item])) #2(#9(#3(value_range) #4([value_range] [qualified_representation_item])) #8(shape_dimension_representation)) #7(#3(value_representation_item) #4([value_representation_item] [qualified_representation_item]))	45 45 45 214 214 45 47 43 43 45		#2({value_range <= compound_representation_item})

#5: S1#UNIT#1:: #6: S1#UNIT#2:: #7: If the value_with_unit is a numerical_value or a value_limit and the unit is specified globally.:: #8: If the value_with_unit is a value_range referenced by a Geometric_tolerance.:: #9: If the value_with_unit is not referenced by a Geometric_tolerance.::				
significant_digits	precision_qualifier.precision_value	45	38	#9(qualified_representation_item qualified_representation_item.qualifiers[i] ->value_qualifier value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value) #8(shape_dimension_representation <= shape_representation <= representation representation.items[i] ->representation_item => measure_representation_item <= measure_with_unit <- measure_qualification.qualifiers[1] ->value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value)
value_with_unit to unit (as unit_component)	PATH			#1(measure_representation_item <=) #2(#9(compound_representation_item compound_representation_item.item_element ->compound_item_definition compound_item_definition = set_representation_item set_representation_item set_representation_item[i] ->) #8(shape_dimension_representation <= shape_representation <=

				representation representation.items[i] ->) representation_item => measure_representation_item <=) measure_with_unit measure_with_unit.unit_component -> unit #5(unit = named_unit named_unit) #6(unit = derived_unit derived_unit)
PROPERTY_VALUE #1: If property_value is a value_list.:: #2: If property_value is a string_value.:: #3: If property_value is a numerical_value or a value_limit and the unit is not assigned globally.:: #4: If property_value is a value_range.:: #5: If no significant_digits are given for the value_with_unit.:: #6: If significant_digits are given for the value_with_unit.:: #7: If the property_value is a numerical_value or a value_limit and the unit is specified globally.:: #8: If the value_with_unit is a value_range referenced by a Geometric_tolerance.::	#1(compound_representation_item) #2(descriptive_representation_item) #3(#5(measure_representation_item) #6([measure_representation_item] [qualified_representation_item])) #4(#5(value_range) #6([value_range] [qualified_representation_item])) #7(#5(value_representation_item) #6([value_representation_item] [qualified_representation_item])) #8(shape_dimension_representation)	*43* *45* *45* *45* *45* *214* *214* *45* *43* *43* *45* 47		#2(descriptive_representation_item {descriptive_representation_item <= representation_item => qualified_representation_item qualified_representation_item.qualifiers[1] -> value_qualifier value_qualifier = type_qualifier type_qualifier type_qualifier.name type_qualifier.name = 'string'}) #4(#5(value_range <= compound_representation_item) #6([value_range <= compound_representation_item] [qualified_representation_item]))
value_name	representation_item.name	43		#1(compound_representation_item <=) #2(descriptive_representation_item <=) #3(measure_representation_item <=) #4(value_range <= compound_representation_item <=) #7(value_representation_item <=) #8(shape_dimension_representation<=) representation_item

				representation_item.name

4.1.2 Value_limitation

- Adopt AP224 mapping both for inside and outside of a dimension
- Change to OR case mapping
- Value_limitation.envelope:
  - As attribute envelope is moved from Value_limitation to Size_dimension, mapping has to be moved, too
  - Replace current mapping by AP224 approach

VALUE_LIMITATION #1 : If the limited_value is for a dimension_value.:: #2: If the limited_value is for a any other value.::	#1(plus_minus_tolerance) #2(qualified_representation_item)	47 45		
~~envelope~~	~~applied_classification_assignment~~	~~214~~	~~13~~	~~plus_minus_tolerance classification_item = plus_minus_tolerance classification_item <- applied_classification_assignment.items[i] applied_classification_assignment <= classification_assignment {[classification_assignment.role ->~~ ~~classification_role classification_role.name = 'further tolerance requirement'] [classification_assignment.assigned_class ->~~ ~~group {[group.name = 'E'] [group.description = 'dimensioning principle']}]}~~
value_limitation to limits_and_fits (as is_defined_by)	#1(PATH) #2(IDENTICAL MAPPING)			#1(plus_minus_tolerance plus_minus_tolerance.range -> tolerance_method_definition tolerance_method_definition = limits_and_fits limits_and_fits)

value_limitation to plus_minus_bounds (as is_defined_by)	#1(PATH) #2(IDENTICAL MAPPING)			#1(plus_minus_tolerance plus_minus_tolerance.range ->) tolerance_method_definition tolerance_method_definition = tolerance_value tolerance_value)
value_limitation to numerical_value (as limited_value)	#1(PATH) #2([measure_representation_item] [qualified_representation_item])			#1(plus_minus_tolerance plus_minus_tolerance.toleranced_dimension ->) ~~dimensional_characteristic~~ ~~dimensional_characteristic = dimensional_size~~ ~~dimensional_size~~ ~~dimensional_characteristic = dimensional_size~~ dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation ->) shape_dimension_representation <= shape_representation <= representation representation.items[i] ->) representation_item => measure_representation_item)

PLUS_MINUS_BOUNDS #1 : If the plus_minus_bounds define a value_limitation for a dimension_value.:: #2: If the plus_minus_bounds define a value_limitation for any other value.::	#1(tolerance_value) #2(qualified_representation_item)	47 45	48	
lower_bound	#1(tolerance_value.lower_bound) #2(standard_uncertainty.uncertainty_value)	47 47		#2({qualified_representation_item <= representation_item} qualified_representation_item qualified_representation_item.qualifiers[i] ->

				value_qualifier value_qualifier = uncertainty_qualifier uncertainty_qualifier.measure_name='lower limit' uncertainty_qualifier => standard_uncertainty standard_uncertainty.uncertainty_value)
significant_digits	precision_qualifier.precision_value	45	38	#1([tolerance_value tolerance_value.lower_bound -> measure_with_unit <-] [tolerance_value tolerance_value.upper_bound -> measure_with_unit <-] measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] ->) #2(qualified_representation_item qualified_representation_item.qualifiers[i] ->) value_qualifier value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value
upper_bound	#1(tolerance_value.upper_bound) #2(standard_uncertainty.uncertainty_value)	47 47		#2({qualified_representation_item <= representation_item} qualified_representation_item qualified_representation_item.qualifiers[i] ->> value_qualifier value_qualifier = uncertainty_qualifier uncertainty_qualifier.measure_name='upper limit' uncertainty_qualifier => standard_uncertainty

				standard_uncertainty.uncertainty_value)
value_determination	type_qualifier.name	45	49	#1(tolerance_value [tolerance_value.lower_bound ->] [tolerance_value.upper_bound->] measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[i] ->) #2(qualified_representation_item qualified_representation_item.qualifiers[i] ->) value_qualifier value_qualifier = type_qualifier type_qualifier type_qualifier.name {(type_qualifier.name) (type_qualifier.name = 'required') (type_qualifier.name = 'designed') (type_qualifier.name = 'calculated') (type_qualifier.name = 'measured') (type_qualifier.name = 'estimated')}

LIMITS_AND_FITS #1 : If the limits_and_fits define a value_limitation for a dimension_value.:: #2: If the limits_and_fits define a value_limitation for any other value.::	#1(limits_and_fits) #2(qualified_representation_item)	47 45		
deviation	#1(limits_and_fits.form_variance) #2(qualitative_uncertainty.uncertainty_value)	47 45		#2({qualified_representation_item <= representation_item} qualified_representation_item qualified_representation_item.qualifiers[i] ->) value_qualifier

				value_qualifier = uncertainty_qualifier uncertainty_qualifier.measure_name='form variance' uncertainty_qualifier => qualitative_uncertainty qualitative_uncertainty.uncertainty_value)
fitting_type	#1(limits_and_fits.zone_variance) #2(qualitative_uncertainty.uncertainty_value)	47 45		#2({qualified_representation_item <= representation_item} qualified_representation_item qualified_representation_item.qualifiers[i] ->> value_qualifier value_qualifier = uncertainty_qualifier uncertainty_qualifier.measure_name='zone variance' uncertainty_qualifier => qualitative_uncertainty qualitative_uncertainty.uncertainty_value)
grade	#1(limits_and_fits.grade) #2(qualitative_uncertainty.uncertainty_value)	47 45		#2({qualified_representation_item <= representation_item} qualified_representation_item qualified_representation_item.qualifiers[i] ->> value_qualifier value_qualifier = uncertainty_qualifier uncertainty_qualifier.measure_name='grade' uncertainty_qualifier => qualitative_uncertainty qualitative_uncertainty.uncertainty_value)

SIZE_DIMENSION #1 : If the envelope principle applies.::	dimensional_size	47	22	

#2: If the envelope principle does not apply.::				
size_dimension to annotation_curve (as is_applied_to)	PATH			dimensional_size dimensional_size.applies_to -> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition property_definition.definition property_definition represented_definition = property_definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence
size_dimension to shape_element (as is_applied_to)	PATH			dimensional_size dimensional_size.applies_to -> shape_aspect
envelope	#1(representation.name) #2(NO MAPPING)	47		#1(dimensional_size dimensional_characteristic = dimensional_size dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.name {representation.name = 'envelope tolerance'})

4.1.3 General_size_dimension

- Rename to Externally_defined_size_dimension

~~GENERAL~~EXTERNALLY_DEFINED_SIZE_DIMENSION #1: If no measurement_path is specified.:: #2: If a measurement_path is specified.::	#1(~~dimensional_size~~ externally_defined_dimension_definition) #2([externally_defined_dimension_definition dimensional_size_with_path])	~~47~~214 47,214	22	externally_defined_dimension_definition <= externally_defined_item { externally_defined_item.item_id ->>source_item source_item = 'external size dimension'} { externally_defined_item.source ->>external_source external_source.source_id ->>source_item source_item = 'external size dimension specification' }
~~dimension_type~~	~~dimensional_size.name~~	~~47~~		~~#1(dimensional_size)~~ ~~#2(dimensional_size_with_path <= dimensional_size)~~ ~~dimensional_size.name~~
name	Dimensional_size.name	47		externally_defined_dimension_definition <= dimensional_size ->>dimensional_size.name
~~GENERAL~~EXTERNALLY_DEFINED_size_dimension to measurement_path (as used_path)	PATH			dimensional_size_with_path dimensional_size_with_path.path ->>shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <-

				property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation

4.1.4 Angular_size_dimension

- Angular_size_dimension.full: change mapping as proposed

ANGULAR_SIZE_DIMENSION #1: If the angle is established between the two sides of the angular element.:: #2: If the angle is established between the bisector line and the angular element.::	angular_size	47	22	
full	~~dimensional_size.name~~ representation_item.name	47 43		~~angular_size <=~~ ~~dimensional_size~~ ~~{#1(dimensional_size.name = 'full angle dimension')~~ ~~#2(dimensional_size.name = 'half angle dimension')}~~ angular_size <= dimensional_size dimensional_characteristic = dimensional_size dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation ->shape_dimension_representation <= shape_representation <= representation representation.items[i] ->{ representation_item.name = 'angle type' } representation_item => descriptive_representation_item descriptive_representation_item.description {#1:(descriptive_representation_item.description = 'full angle')

				(#2: descriptive_representation_item.description = 'half angle')}
major_angle	angular_size.angle_selection	47		
CURVED_DISTANCE_DIMENSION #1: If the with_curve_direction is 'false'.:: #2: If the with_curve_direction is 'true'.::	#1(dimensional_location_with_path) #2([dimensional_location_with_path] [directed_dimensional_location])	47 47 214		#1(dimensional_location_with_path <=) #2(directed_dimensional_location <= dimensional_location => dimensional_location_with_path <=) dimensional_location <= shape_aspect_relationship {shape_aspect_relationship.name = 'curved distance'}
with_curve_direction	IDENTICAL MAPPING			
curved_distance_dimension to measurement_path (as used_path)	PATH			dimensional_location_with_path dimensional_location_with_path.path -> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation

4.1.5 Linear_distance_dimension

- Move attribute directed to Location_dimension

LINEAR_DISTANCE_DIMENSION #1: If the directed is 'false' ~~and the orientation is~~	#1(dimensional_location) #2(directed_dimensional_location)	47 214		#1(dimensional_location <=) #2(directed_dimensional_location <=

~~not specified.::~~ #2: If the directed is 'true' ~~and the orientation is not specified.::~~ ~~#3: If the directed is 'false' and the orientation is specified.::~~ ~~#4: If the directed is 'true' and the orientation is specified.::~~	~~#3(dimensional_location_with_path)~~ ~~#4([directed_dimensional_location]~~ ~~[dimensional_location_with_path])~~	~~47~~ ~~214~~ ~~47~~		dimensional_location <=) ~~#3(dimensional_location_with_path <=~~ ~~dimensional_location <=)~~ ~~#4(directed_dimensional_location <=~~ ~~dimensional_location =>~~ ~~dimensional_location_with_path <=~~ ~~dimensional_location <=)~~ shape_aspect_relationship {shape_aspect_relationship.name = 'linear distance'}
~~directed~~	~~IDENTICAL MAPPING~~			
orientation	placement	42		dimensional_location_with_path <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement => (axis2_placement_2d) (axis2_placement_2d)

4.1.6 Location_dimension

LOCATION_DIMENSION ~~#1: If it is not an angular_location_dimension or if it is an angular_location_dimension whose origin_determination is 'unchanged'.::~~ ~~#2: If it is an angular_location_dimension whose origin_determination is not 'unchanged'.::~~ ~~#3: If it is an angular_location_dimension whose origin_determination is 'positive normal' or 'negative normal'.::~~ ~~#4: If it is an angular_location_dimension whose origin_determination is 'mirror'.::~~ ~~#5: If it is an angular_location_dimension whose origin_determination is 'positive normal'.::~~ ~~#6: If it is an angular_location_dimension whose origin_determination is 'negative normal'.::~~ ~~#7: If it is not an angular_location_dimension or if it is an angular_location_dimension whose mirror_target is 'false'.::~~ ~~#8: If it is an angular_location_dimension whose mirror_target is 'true'.::~~ #1: If the directed is 'false' #2: If the directed is 'true' #9: For the primary_language_dependent_string.:: #10: For any additional_language_dependent_string.::	#1(dimensional_location) #2(directed_dimensional_location)	47 47		({directed_dimensional_location <= dimensional_location})
directed	IDENTICAL MAPPING			

description	shape_aspect_relationship.description	41		dimensional_location <= shape_aspect_relationship shape_aspect_relationship.description
location_dimension to multi_language_string (as description)	#9([shape_aspect_relationship.description] [PATH]) #10(PATH)	41	165	dimensional_location <= shape_aspect_relationship #9(attribute_language_item = shape_aspect_relationship attribute_language_item <- attribute_language_assignment.items[i] attribute_language_assignment {attribute_language_assignment <= attribute_classification_assignment attribute_classification_assignment.attribute_name = 'description'}) #10(multi_language_attribute_item = shape_aspect_relationship multi_language_attribute_item <- multi_language_attribute_assignment.items[i] multi_language_attribute_assignment {multi_language_attribute_assignment <= attribute_value_assignment attribute_value_assignment.attribute_name = 'description'})
location_dimension to annotation_curve (as origin)	PATH			dimensional_location <= shape_aspect_relationship ~~#1(~~shape_aspect_relationship.relating_shape_aspect ->~~)~~ ~~#2(shape_aspect_relationship.relating_shape_aspect ->~~ ~~shape_aspect <-~~ ~~{shape_aspect =>~~ ~~derived_shape_aspect =>~~ ~~#3(perpendicular_to)~~ ~~#4(extension)}~~ ~~shape_aspect_relationship.relating_shape_aspect~~ ~~shape_aspect_relationship~~ ~~{shape_aspect_relationship =>~~ ~~shape_aspect_deriving_relationship}~~ ~~shape_aspect_relationship.related_shape_aspect ->)~~

				shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence
location_dimension to constructive_geometric_element (as origin)	PATH			dimensional_location <= shape_aspect_relationship ~~#1(~~shape_aspect_relationship.relating_shape_aspect ->~~)~~ ~~#2(shape_aspect_relationship.relating_shape_aspect ->~~ ~~shape_aspect <-~~ ~~(shape_aspect =>~~ ~~derived_shape_aspect =>~~ ~~#3(perpendicular_to)~~ ~~#4(extension))~~ ~~shape_aspect_relationship.relating_shape_aspect~~ ~~shape_aspect_relationship~~ ~~(shape_aspect_relationship =>~~ ~~shape_aspect_deriving_relationship)~~ ~~shape_aspect_relationship.related_shape_aspect ->)~~ shape_aspect shape_definition = shape_aspect shape_definition

				characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation representation.items[i] ->representation_item
location_dimension to derived_geometry (as origin)	PATH			dimensional_location <= shape_aspect_relationship ~~#1(~~shape_aspect_relationship.relating_shape_aspect ->~~)~~ ~~#2(shape_aspect_relationship.relating_shape_aspect -> shape_aspect <- {shape_aspect => derived_shape_aspect => #3(perpendicular_to) #4(extension)} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship => shape_aspect_deriving_relationship} shape_aspect_relationship.related_shape_aspect ->)~~ shape_aspect => derived_shape_aspect
location_dimension to shape_element (as origin)	PATH			dimensional_location <= shape_aspect_relationship ~~#1(~~shape_aspect_relationship.relating_shape_aspect ->~~)~~ ~~#2(shape_aspect_relationship.relating_shape_aspect -> shape_aspect <- {shape_aspect =>~~

				~~derived_shape_aspect =>~~ ~~#3(perpendicular_to)~~ ~~#4(extension)]~~ ~~shape_aspect_relationship.relating_shape_aspect~~ ~~shape_aspect_relationship~~ ~~[[shape_aspect_relationship =>~~ ~~shape_aspect_deriving_relationship]~~ ~~[#5(shape_aspect_relationship.name = 'positive normal relationship')~~ ~~#6(shape_aspect_relationship.name = 'negative normal relationship')~~ ~~#4(shape_aspect_relationship.name = 'mirroring relationship')]]~~ ~~shape_aspect_relationship.related_shape_aspect ->)~~ shape_aspect
location_dimension to annotation_curve (as target)	PATH			dimensional_location <= shape_aspect_relationship ~~#7(~~shape_aspect_relationship.related_shape_aspect ->~~)~~ ~~#8(shape_aspect_relationship.related_shape_aspect ->~~ ~~shape_aspect <-~~ ~~[shape_aspect =>~~ ~~derived_shape_aspect =>~~ ~~extension]~~ ~~shape_aspect_relationship.relating_shape_aspect~~ ~~shape_aspect_relationship~~ ~~[[shape_aspect_relationship =>~~ ~~shape_aspect_deriving_relationship]~~ ~~[shape_aspect_relationship.name = 'mirroring relationship']]~~ ~~shape_aspect_relationship.related_shape_aspect ->)~~ shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition

				property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence
location_dimension to constructive_geometric_element (as target)	PATH			dimensional_location <= shape_aspect_relationship ~~#7(~~shape_aspect_relationship.related_shape_aspect ->~~)~~ ~~#8(shape_aspect_relationship.related_shape_aspect ->~~ ~~shape_aspect <-~~ ~~{shape_aspect =>~~ ~~derived_shape_aspect =>~~ ~~extension}~~ ~~shape_aspect_relationship.relating_shape_aspect~~ ~~shape_aspect_relationship~~ ~~{[shape_aspect_relationship =>~~ ~~shape_aspect_deriving_relationship]~~ ~~[shape_aspect_relationship.name = 'mirroring relationship']}~~ ~~shape_aspect_relationship.related_shape_aspect ->)~~ shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <-

				property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation representation.items[i] ->representation_item
location_dimension to derived_geometry (as target)	PATH			dimensional_location <= shape_aspect_relationship ~~#7(~~shape_aspect_relationship.related_shape_aspect ->~~)~~ ~~#8(shape_aspect_relationship.related_shape_aspect ->~~ ~~shape_aspect <-~~ ~~{shape_aspect =>~~ ~~derived_shape_aspect =>~~ ~~extension)~~ ~~shape_aspect_relationship.relating_shape_aspect~~ ~~shape_aspect_relationship~~ ~~{[shape_aspect_relationship =>~~ ~~shape_aspect_deriving_relationship]~~ ~~[shape_aspect_relationship.name = 'mirroring relationship']}~~ ~~shape_aspect_relationship.related_shape_aspect ->)~~ shape_aspect => derived_shape_aspect
location_dimension to shape_element (as target)	PATH			dimensional_location <= shape_aspect_relationship ~~#7(~~shape_aspect_relationship.related_shape_aspect ->~~)~~ ~~#8(shape_aspect_relationship.related_shape_aspect ->~~ ~~shape_aspect <-~~ ~~{shape_aspect =>~~ ~~derived_shape_aspect =>~~ ~~extension)~~ ~~shape_aspect_relationship.relating_shape_aspect~~ ~~shape_aspect_relationship~~ ~~{[shape_aspect_relationship =>~~ ~~shape_aspect_deriving_relationship]~~

				~~[shape_aspect_relationship.name = 'mirroring relationship']~~ ~~shape_aspect_relationship.related_shape_aspect ->)~~ shape_aspect

4.1.7 Angular_location_dimension

- Delete mappings for major_angle, mirror_target, origin_termination
- Add mapping for orientation

ANGULAR_LOCATION_DIMENSION ~~#1: If the origin_determination is 'unchanged'.::~~ ~~#2: If the origin_determination is not 'unchanged'.::~~ ~~#3: If the origin_determination is 'positive normal' or 'negative normal'.::~~ ~~#4: If the origin_determination is 'mirror'.::~~ ~~#5: If the origin_determination is 'positive normal'.::~~ ~~#6: If the origin_determination is 'negative normal'.::~~ ~~#7: If the mirror_target is 'false'.::~~ ~~#8: If the mirror_target is 'true'.::~~	angular_location	47		
~~major_angle~~	~~angular_location.angle_selection~~	~~47~~		
~~mirror_target~~	~~#7(shape_aspect)~~ ~~#8(shape_aspect_deriving_relationship)~~	~~41~~ ~~47~~		~~angular_location <=~~ ~~dimensional_location <=~~ ~~shape_aspect_relationship~~ ~~#7(shape_aspect_relationship.related_shape_aspect ->~~ ~~shape_aspect)~~ ~~#8(shape_aspect_relationship.related_shape_aspect ->~~ ~~shape_aspect <-~~ ~~(shape_aspect =>~~ ~~derived_shape_aspect =>~~ ~~extension)~~

				shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship => {shape_aspect_relationship.name = 'mirroring relationship'} shape_aspect_deriving_relationship)
origin_determination	#1(shape_aspect) #2(shape_aspect_deriving_relationship)	41 47		angular_location <= dimensional_location <= shape_aspect_relationship #1(shape_aspect_relationship.relating_shape_aspect ->shape_aspect) #2(shape_aspect_relationship.relating_shape_aspect ->shape_aspect <- {shape_aspect => derived_shape_aspect => #3(perpendicular_to) #4(extension)} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship => {#5(shape_aspect_relationship.name = 'positive normal relationship') #6(shape_aspect_relationship.name = 'negative normal relationship') #4(shape_aspect_relationship.name = 'mirroring relationship')} shape_aspect_deriving_relationship)
orientation	placement			angular_location <= dimensional_location <= shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation

				property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement

4.1.8 Geometric_dimension

- Add mapping for attribute dimension_note

GEOMETRIC_DIMENSION #1: If the geometric_dimension is a location_dimension.:: #2: If the geometric_dimension is a size_dimension.:: #3: If the value_with_unit is a numerical_value or a value_limit and the unit is not assigned globally.:: #4: If the value_with_unit is a value_range.::	#1(dimensional_location) #2(dimensional_size)	47 47	22	
id	#1(shape_aspect_relationship.name) #2(identification_assignment.assigned_id)	41 41	29	#1(dimensional_location <= shape_aspect_relationship shape_aspect_relationship.name) #2(dimensional_size identification_item = dimensional_size identification_item <- applied_identification_assignment.items[i] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'size id'}

				identification_assignment.assigned_id)
geometric_dimension to value_with_unit (as dimension_value)	PATH			#1(dimensional_location dimensional_characteristic = dimensional_location) #2(dimensional_size dimensional_characteristic = dimensional_size) dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items [i] -> representation_item => #3(measure_representation_item) #4(measure_representation_item {measure_representation_item <= representation_item <- set_representation_item [i] set_representation_item compound_item_definition = set_representation_item compound_item_definition <- compound_representation_item.item_element compound_representation_item => value_range})
notes	Descriptive_representation_item.description			#1(dimensional_location dimensional_characteristic = dimensional_location) #2(dimensional_size dimensional_characteristic = dimensional_size) dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <=

				shape_representation <= representation representation.items[i] ->> representation_item {representation_item.name = 'dimensional note'} representation_item => descriptive_representation_item descriptive_representation_item.description {(descriptive_representation_item.description) (descriptive_representation_item.description = ‘auxiliary’) (descriptive_representation_item.description = ‘theoretical’)}

4.1.9 Document_assignment

- Add new role value for the specification of dimensioning standards
- Add new association to Externally_defined_size_dimension

DOCUMENT_ASSIGNMENT *No change in OR case description!!*	#1(applied_document_reference) #2(applied_document_usage_constraint_assignment)	214 214		#1(applied_document_reference <= document_reference) #2(applied_document_usage_constraint_assignment <= document_usage_constraint_assignment)
role	#1(object_role.name) #2(document_usage_role.name)	41 41	23, 32	#1(applied_document_reference <= document_reference document_reference.role ->> object_role object_role.name {(object_role.name) (object_role.name = 'additional information') (object_role.name = 'behavior') (object_role.name = 'catalogue') (object_role.name = 'description') (object_role.name = 'informative') (object_role.name = 'mandatory') (object_role.name = 'manual')

				(object_role.name = 'mathematical description') (object_role.name = 'specification') (object_role.name = 'dimensioning standard')}) #2(applied_document_usage_constraint_assignment <= document_usage_constraint_assignment document_usage_constraint_assignment.role ->) document_usage_role document_usage_role.name {(document_usage_role.name) (document_usage_role.name = 'additional information') (document_usage_role.name = 'behavior') (document_usage_role.name = 'catalogue') (document_usage_role.name = 'description') (document_usage_role.name = 'informative') (document_usage_role.name = 'mandatory') (document_usage_role.name = 'manual') (document_usage_role.name = 'mathematical description') (document_usage_role.name = 'specification') (object_role.name = 'dimensioning standard')})
document_assignment to Externally_defined_size_dimension (as is_assigned_to)	PATH			#1(applied_document_reference applied_document_reference.items[i] ->) #2(applied_document_usage_constraint_assignment applied_document_usage_constraint_assignment.items[i] ->) document_reference_item document_reference_item = externally_defined_dimension_definition externally_defined_dimension_definition

4.1.10 Datum

- Move attribute precedence to new Datum_reference

DATUM	datum~~_reference~~	47		
~~precedence~~	~~datum_reference.precedence~~	~~47~~		~~datum_reference~~ ~~datum_reference.precedence~~

4.1.11 Single_datum

SINGLE_DATUM	datum	47		
datum_name	datum.identification	47		
~~single_datum to datum_target_set~~ ~~(as is_defined_by)~~	~~IDENTICAL MAPPING~~			
~~single_datum to derived_geometry~~ ~~(as is_defined_by)~~	~~IDENTICAL MAPPING~~			~~datum <=~~ ~~shape_aspect =>~~ ~~derived_shape_aspect~~
~~single_datum to shape_element~~ ~~(as is_defined_by)~~	~~PATH~~			~~datum <=~~ ~~shape_aspect <-~~ ~~shape_aspect_relationship.related_shape_aspect~~ ~~shape_aspect_relationship~~ ~~shape_aspect_relationship.relating_shape_aspect ->~~ ~~shape_aspect~~ ~~{shape_aspect =>~~ ~~datum_feature}~~
single_datum to tolerance_condition (as modification)	referenced_modified_datum.modifier	47		datum <- datum_reference.referenced_datum datum_reference datum_reference => referenced_modified_datum referenced_modified_datum.modifier

4.1.12 References to Datum

- Replace the mapping of attribute reference_datum for
  - perpendicularity_tolerance
  - parallelism_tolerance
  - angularity_tolerance
  - concentricity_tolerance
  - symmetry_tolerance
  - position_tolerance
  - circular_runout_tolerance
  - total_runout_tolerance
  - line_profile_tolerance
  - surface_profile_tolerance
  - coaxiality_tolerance
  -

xxx_tolerance to datum_reference (as reference_datum)	PATH	xxx_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference

4.2 New Mappings

4.2.1 Curved_size_dimension

CURVED_SIZE_DIMENSION	dimensional_size	47		{dimensional_size dimensional_size.name = 'curve length'}

4.2.2 Diameter_size_dimension

DIAMETER_SIZE_DIMENSION	dimensional_size	47		{dimensional_size

				dimensional_size.name = ‘diameter’}

4.2.3 Height_size_dimension

HEIGHT_SIZE_DIMENSION #1: If no measurement_path is specified.:: #2: If a measurement_path is specified.::	#1(dimensional_size) #2(dimensional_size_with_path)	47 47		#1({dimensional_size dimensional_size.name = ‘height’}) #2({dimensional_size_with_path <= dimensional_size dimensional_size.name = 'height'})
height_size_dimension to measurement_path (as used_path)	PATH			dimensional_size_with_path dimensional_size_with_path.path -> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation

4.2.4 Length_size_dimension

LENGTH_SIZE_DIMENSION #1: If no measurement_path is specified.:: #2: If a measurement_path is specified.::	#1(dimensional_size) #2(dimensional_size_with_path)	47 47		#1({dimensional_size dimensional_size.name = ‘length’}) #2({dimensional_size_with_path <= dimensional_size dimensional_size.name = ‘length’})

length_size_dimension to measurement_path (as used_path)	PATH			dimensional_size_with_path dimensional_size_with_path.path -> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation

4.2.5 Radial_size_dimension

RADIAL_SIZE_DIMENSION #1: If the centre of the radium is located.:: #2: If the radius is located by its end points.::	dimensional_size	47		{dimensional_size dimensional_size.name = ‘radius’}
radius_type	descriptive_representation_item.description	45		dimensional_size dimensional_characteristic = dimensional_size dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items[i] -> {representation_item.name = 'radius type'} representation_item =>

				descriptive_representation_item {#1:(descriptive_representation_item.description = 'centred') #2:(descriptive_representation_item.description = 'adjoining')}

4.2.6 *Width_size_dimension*

WIDTH_SIZE_DIMENSION #1: If no measurement_path is specified.:: #2: If a measurement_path is specified.::	#1(dimensional_size) #2(dimensional_size_with_path)	47 47		#1({dimensional_size dimensional_size.name = 'width'}) #2({dimensional_size_with_path <= dimensional_size dimensional_size.name = 'width'})
width_size_dimension to measurement_path (as used_path)	PATH			dimensional_size_with_path dimensional_size_with_path.path -> shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation

4.2.7 *Datum_reference*

DATUM_REFERENCE	datum_reference	47		
precedence	datum_reference.precedence	47		

Datum_reference to Datum (as referenced_datum)	PATH			datum_reference datum_reference.referenced_datum -> datum

4.2.8 Datum_defined_by_derived_shape

DATUM_DEFINED_BY_DERIVED_SHAPE	datum	47		{datum <= shape_aspect => derived_shape_aspect}

4.2.9 Datum_defined_by_feature

DATUM_DEFINED_BY_FEATURE	datum	47		
Datum_defined_by_feature to Shape_element (as defined_by)	PATH			datum <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => datum_feature}

4.2.10 Datum_defined_by_targets

DATUM_DEFINED_BY_TARGETS	datum	47		
rule_description	rule_description			datum <= shape_aspect shape_aspect.description
Datum_defined_by_targets to	#1([shape_aspect.description]	41	164	datum_ <=

multi_language_string (as rule_description)	[PATH]) #2(PATH)			shape_aspect #1(attribute_language_item = shape_aspect attribute_language_item <- attribute_language_assignment.items[i] attribute_language_assignment {attribute_language_assignment <= attribute_classification_assignment attribute_classification_assignment.attribute_name = 'description'}) #2(multi_language_attribute_item = shape_aspect multi_language_attribute_item <- multi_language_attribute_assignment.items[i] multi_language_attribute_assignment {multi_language_attribute_assignment <= attribute_value_assignment attribute_value_assignment.attribute_name = 'description'})
Datum_defined_by_targets to datum_target (as defined_by)	PATH			datum <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->shape_aspect => datum_target

4.3 Deleted Mappings

4.3.1 Datum_target_set

5. Clause 5.2 AIM EXPRESS Short Listing

5.1 Modified AIM types

5.1.1 Document_reference_item

- Add externally_defined_dimension_definition

5.2 New AIM entities

5.2.1 Externally_defined_dimension_definition

- Define common subtype of dimensional_size and externally_defined_item

5.2.3.x externally_defined_dimension_definition

An **externally_defined_dimension_definition** is a type of **dimensional_size** and **externally_defined_item** that enables the specification of a kind of dimension that is not defined in this standard.

```
EXPRESS specification:
*)
ENTITY externally_defined_dimension_definition
  SUBTYPE OF (dimensional_size, externally_defined_item);
WHERE
  WR1: (SELF\externally_defined_item.item_id = 'external size
dimension') AND (SELF\externally_defined_item.source.source_id =
'external size dimension specification');
  WR2: 1 >= SIZEOF(QUERY ( adr <* USEDIN(SELF,
'AUTOMOTIVE_DESIGN.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
(adr.assigned_document.description = 'external size dimension
specification') ));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **item_id** shall be 'external size dimension' and the **source** attribute shall reference an **external_source** with a **source_id** of 'externally defined dimension specification'.

WR2: The **externally_defined_dimension_definition** shall be referenced by the **items** attribute of type **applied_document_reference** that references through the **assigned_document** attribute a **document** with **description** of 'externally defined dimension specification'.

6. Bibliography

- Add reference to ISO 1101