INTERNATIONAL STANDARD ISO 10303-111:2007
TECHNICAL CORRIGENDUM 1

Published 2008-12-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Industrial automation systems and integration — Product data representation and exchange —

Part 111: Integrated application resource: Elements for the procedural modelling of solid shapes

TECHNICAL CORRIGENDUM 1

Systèmes d'automatisation industrielle et intégration — Représentation et échange de données de produits —

Partie 111: Ressources d'application intégrée: Éléments pour la modélisation procédurale des formes solides

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO 10303-111:2007 was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data*.

Introduction

The modifications made to ISO 10303-111:2007 have four main objectives:

a) to correct the reference to the type **non_negative_length_measure**, *which was originally defined in ISO 10303-108 but has now been moved to ISO 10303-41;*

b) to disambiguate references to enumerated values in WHERE rules of the entity **extruded_face_solid_with_trim_conditions**;

c) to correct an invalid initialization assignment to a local variable in the function **compute_total_depth**;

d) to rename the enumerated values of the type **blend_radius_variation_type** *to avoid a name space clash with Edition 2 of AP209.*

The opportunity has also been taken to correct an error in the logic of the function **validate_countersink_radii**.

ICS 25.040.40 **Ref. No. ISO 10303-111:2007/Cor.1:2008(E)**

Modifications to the text and figures of ISO 10303-111:2007

Introduction, pp. vii, viii

Replace Figure 1 on p. viii with the following:

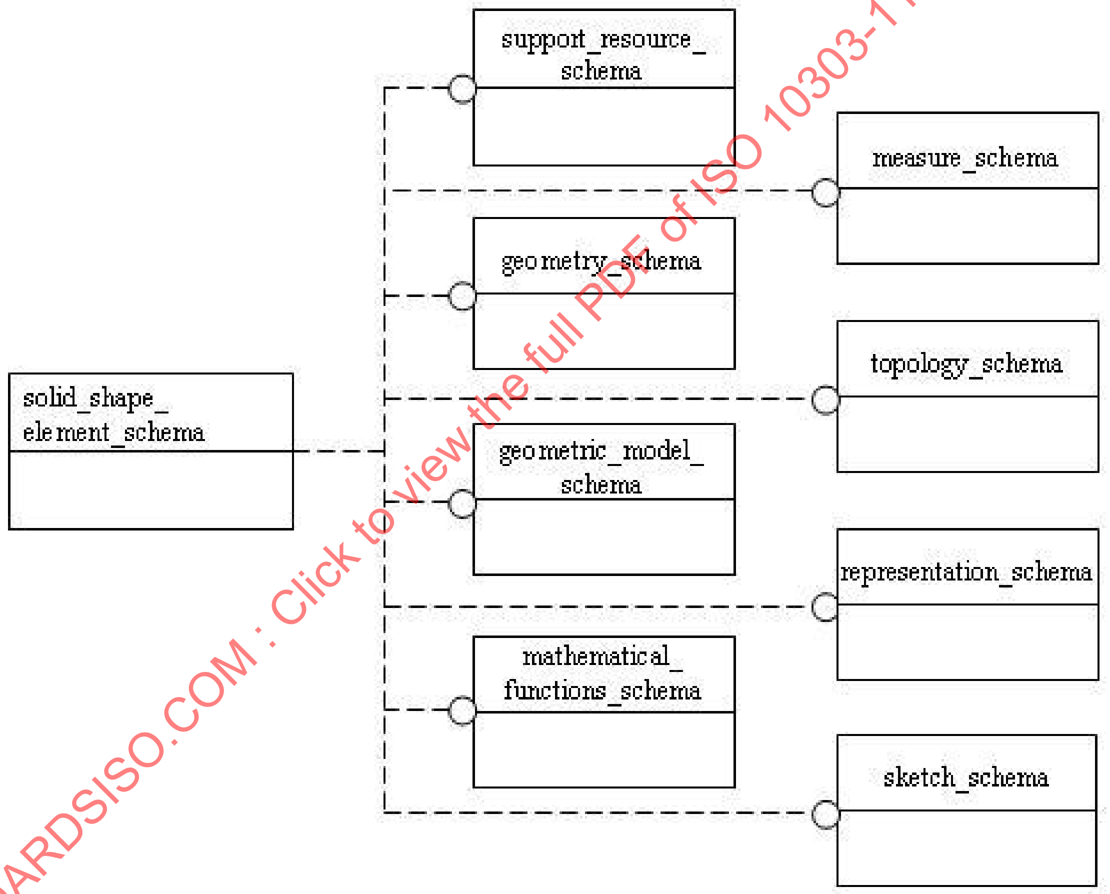


Figure 1 – Schema level diagram of relationships between the solid_-shape_element_schema of this part of ISO 10303 and other resource schemas

Clause 4.1, Introduction, pp. 5,6

Replace the formal reference to the **measure_schema** *on p. 5 with the following:*

```
REFERENCE FROM measure_schema                -- ISO 10303-41
  (length_measure,
   non_negative_length_measure,
   plane_angle_measure,
   positive_length_measure,
   positive_plane_angle_measure);
```

Further, delete the formal reference to the **explicit_geometric_constraint_schema** *on p. 6, and also delete the line specifying that schema from NOTE 1 on the same page.*

Clause 4.3.5, **blend_radius_variation_type***, p. 9*

Replace the EXPRESS code and the enumeration list on p. 9 with the following:

EXPRESS specification:

```
*)
TYPE blend_radius_variation_type = ENUMERATION OF
  (linear_blend,
   cubic_blend,
   unspecified_blend);
END_TYPE;
(*
```

Enumerated item definitions:

linear_blend: the blend radius varies linearly between radius definition points.

cubic_blend: the blend radius varies as a cubic between radius definition points.

unspecified_blend: the blend radius variation is not specified.

Clause 4.4.6, **solid_with_variable_radius_edge_blend***, pp. 17 - 19*

Replace the descriptive text on p. 18 between Notes 3 and 4 with the following:

In any interval whose radius-defining function is **cubic_blend**, the actual function is determined using Hermite interpolation, in terms of the radii at each end point of the interval and values of the first derivatives of the radius variation function at those end points.

Replace the text of Note 5 on p. 18 with the following:

In any interval for which the value of the radius-defining function is **unspecified_blend** it is recommended that linear interpolation is used initially in the receiving system, but that the user is warned that some other native blending capability of that system may be more appropriate.

Clause 4.4.48, **extruded_face_solid_with_trim_conditions***, pp. 64-66*

Replace the EXPRESS code on p. 65 with the following, in which WHERE rules WR2, WR3, and WR4 have been modified:

EXPRESS specification:

```
*)
ENTITY extruded_face_solid_with_trim_conditions
  SUPERTYPE OF (ONEOF
               (extruded_face_solid_with_draft_angle,
                extruded_face_solid_with_multiple_draft_angles))
  SUBTYPE OF (extruded_face_solid);
  first_trim_condition  : trim_condition_select;
  second_trim_condition : trim_condition_select;
  first_trim_intent     : trim_intent;
  second_trim_intent    : trim_intent;
  first_offset          : non_negative_length_measure;
  second_offset         : non_negative_length_measure;
WHERE
  WR1: NOT(('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
         IN TYPEOF(first_trim_condition)) OR
         ('MEASURE_SCHEMA.PLANE_ANGLE_MEASURE'
         IN TYPEOF(second_trim_condition)));
  WR2: NOT ((('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN TYPEOF(first_trim_condition)) AND
         ((first_trim_intent = trim_intent.offset)
         OR (first_trim_intent = trim_intent.up_to_next))) OR
         (('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN TYPEOF(second_trim_condition)) AND
         ((second_trim_intent = trim_intent.offset)
         OR (second_trim_intent = trim_intent.up_to_next))));
  WR3: NOT (((NOT ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN TYPEOF(first_trim_condition))) AND
         ((first_trim_intent = trim_intent.blind)
         OR (first_trim_intent = trim_intent.through_all))) OR
         ((NOT('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN TYPEOF(second_trim_condition))) AND
         ((second_trim_intent = trim_intent.blind)
         OR (second_trim_intent = trim_intent.through_all))));
  WR4: (((first_trim_intent = trim_intent.offset)
           AND (first_offset > 0)) XOR
         ((first_trim_intent <> trim_intent.offset)
           AND (first_offset = 0))) AND
         (((second_trim_intent = trim_intent.offset)
           AND (second_offset > 0)) XOR
         ((second_trim_intent <> trim_intent.offset)
           AND (second_offset = 0)));
  WR5: NOT((('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN TYPEOF(first_trim_condition)) AND
            ('MEASURE_SCHEMA.LENGTH_MEASURE'
         IN TYPEOF(second_trim_condition))) AND
         (first_trim_condition = second_trim_condition));
END_ENTITY;
(*
```

Clause 4.5.2*, compute_total_depth, *pp. 72, 73

Replace the EXPRESS code on p. 72 with the following:

EXPRESS specification:

```
*)
FUNCTION compute_total_depth (swsrh : solid_with_stepped_round_hole)
                                    : positive_length_measure;

LOCAL
i   : positive_integer;
n   : positive_integer := swsrh.segments;
td : positive_length_measure := swsrh.segment_depths[1];
END_LOCAL;

IF n = 1
THEN RETURN(td);
ELSE
  REPEAT i := 2 TO n;
    td := td + swsrh.segment_depths[i];
  END_REPEAT;
END_IF;
RETURN(td);

END_FUNCTION;
(*
```

Clause 4.5.3, **validate_countersink_radii,** ***pp. 73,74***
Replace the EXPRESS code on pp. 73, 74 with the following:

EXPRESS specification:

```
*)
FUNCTION validate_countersink_radii
  (cskhole : solid_with_stepped_round_hole_and_conical_transitions)
           : BOOLEAN;

  LOCAL
    i,j             : INTEGER;
    n               : INTEGER := 1 +
                        cskhole\solid_with_stepped_round_hole.segments;
    smaller, larger : positive_length_measure;
  END_LOCAL;

  REPEAT i := 1 TO SIZEOF(cskhole.conical_transitions);

  -- First check whether transition i applies to the entry of the hole or
  -- the exit of a through hole - those cases only need to be checked for
  -- the sign of the cone apex angle.

  IF (((cskhole.conical_transitions[i].transition_number = 1)
       AND (cskhole.conical_transitions[i].cone_apex_angle < 0))
    XOR ((cskhole.conical_transitions[i].transition_number = n)
         AND (cskhole.conical_transitions[i].cone_apex_angle > 0)))
  THEN RETURN(FALSE);
  ELSE
    IF ((cskhole.conical_transitions[i].transition_number <> 1)
      AND (cskhole.conical_transitions[i].transition_number <> n))
    THEN

  -- For all remaining transitions, check that the cone base radius
  -- lies in the range of validity.

      BEGIN
        j := cskhole.conical_transitions[i].transition_number;
        IF cskhole\solid_with_stepped_round_hole.segment_radii[j]
          > cskhole\solid_with_stepped_round_hole.segment_radii[j-1]
        THEN
          BEGIN
            IF (cskhole.conical_transitions[i].cone_apex_angle > 0)
            THEN RETURN(FALSE);
            END_IF;
            larger
              := cskhole\solid_with_stepped_round_hole.segment_radii[j];
            smaller
              := cskhole\solid_with_stepped_round_hole.segment_radii[j-1];
          END;
        ELSE
          BEGIN
            IF (cskhole.conical_transitions[i].cone_apex_angle < 0)
            THEN RETURN(FALSE);
            END_IF;
            larger
              := cskhole\solid_with_stepped_round_hole.segment_radii[j-1];
            smaller
              := cskhole\solid_with_stepped_round_hole.segment_radii[j];
          END;
        END_IF;
        IF ((cskhole.conical_transitions[i].cone_base_radius > larger)
          OR (cskhole.conical_transitions[i].cone_base_radius < smaller))
        THEN RETURN(FALSE);
        END_IF;
      END;
    END_IF;
  END_IF;
  END_REPEAT;
  RETURN(TRUE);

END_FUNCTION;
(*
```

Annex D, pp. 79ff.
The changes identified in this Technical Corrigendum require five of the EXPRESS-G diagrams to be changed. Replace Figures D.5, D.6, D.7, D.8 and D.10, respectively, with the following diagrams:

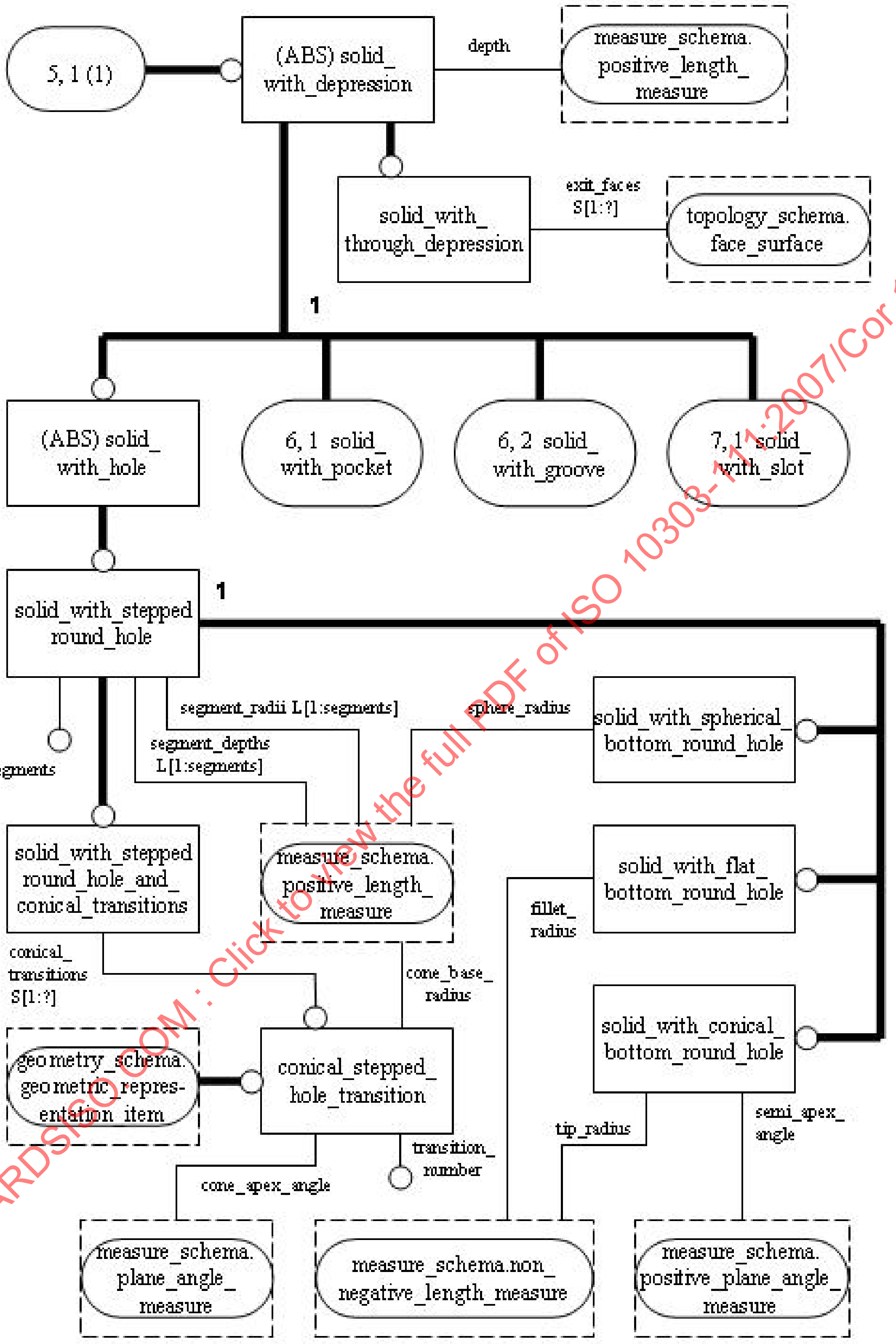
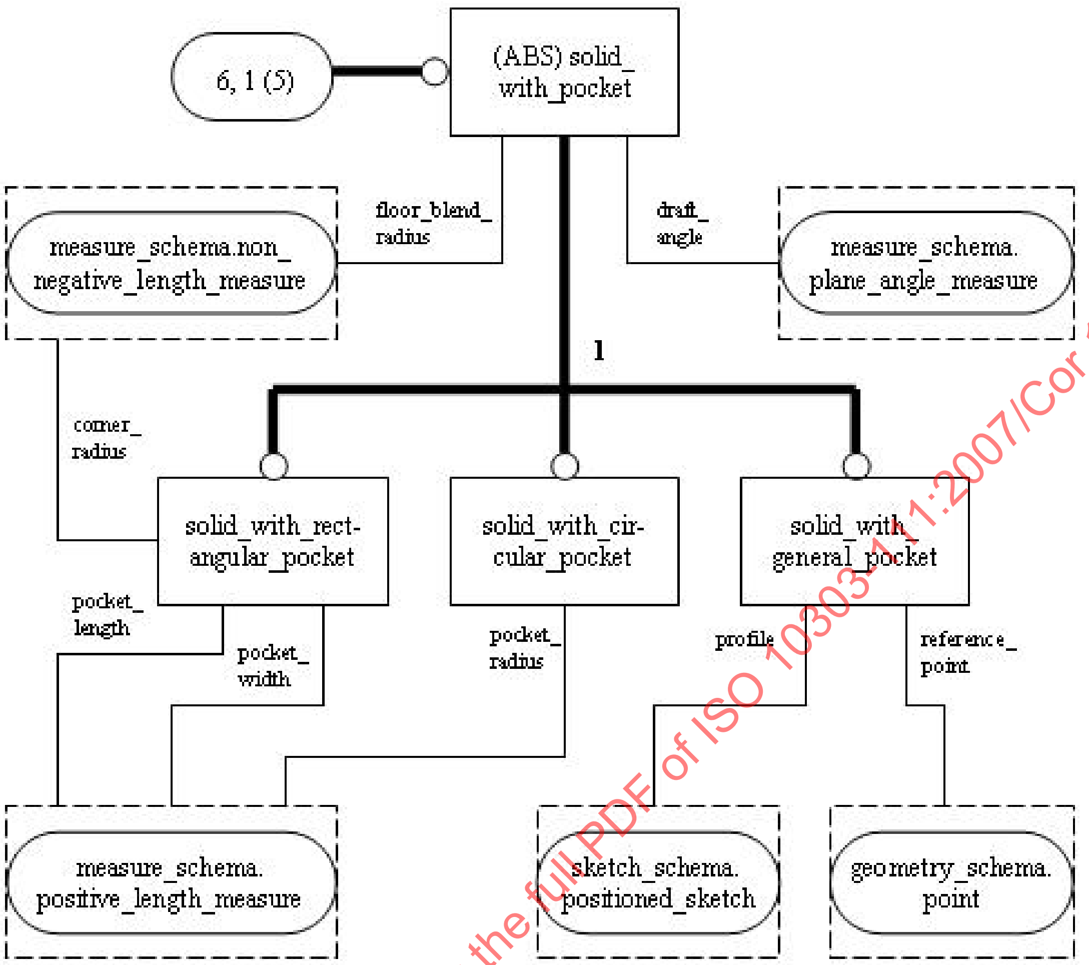

5, 1 (1)
(ABS) solid_ with_depression
depth
measure_schema. positive_length_ measure
solid_with_ through_depression
exit_faces S[1:?]
topology_schema. face_surface
1
(ABS) solid_ with_hole
6, 1 solid_ with_pocket
6, 2 solid_ with_groove
7, 1 solid_ with_slot
solid_with_stepped_ round_hole
1
segments
segment_radii L[1:segments]
segment_depths L[1:segments]
sphere_radius
solid_with_spherical_ bottom_round_hole
measure_schema. positive_length_ measure
solid_with_stepped_ round_hole_and_ conical_transitions
solid_with_flat_ bottom_round_hole
fillet_ radius
conical_ transitions S[1:?]
cone_base_ radius
solid_with_conical_ bottom_round_hole
geometry_schema. geometric_repres- entation_item
conical_stepped_ hole_transition
tip_radius
semi_apex_ angle
transition_ number
cone_apex_angle
measure_schema. plane_angle_ measure
measure_schema.non_ negative_length_measure
measure_schema. positive_plane_angle_ measure

Figure D.5 – solid_shape_element_schema – EXPRESS-G diagram 5 of 11


6, 1 (5)
(ABS) solid_ with_pocket
floor_blend_ radius
draft_ angle
measure_schema.non_ negative_length_measure
measure_schema. plane_angle_measure
1
corner_ radius
solid_with_rect-angular_pocket
solid_with_cir-cular_pocket
solid_with_ general_pocket
pocket_ length
pocket_ width
pocket_ radius
profile
reference_ point
measure_schema. positive_length_measure
sketch_schema. positioned_sketch
geometry_schema. point

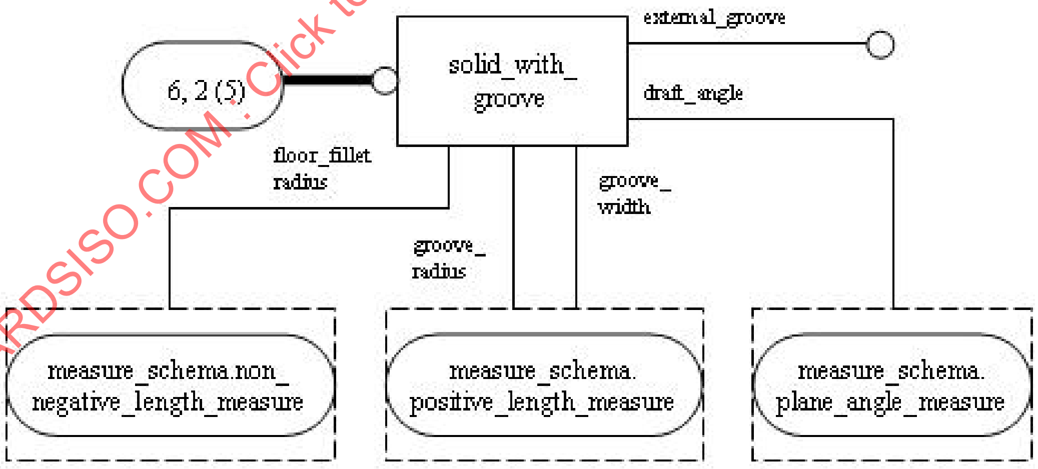

6, 2 (5)
solid_with_ groove
external_groove
draft_angle
floor_fillet radius
groove_ width
groove_ radius
measure_schema.non_ negative_length_measure
measure_schema. positive_length_measure
measure_schema. plane_angle_measure

Figure D.6 – solid_shape_element_schema – EXPRESS-G diagram 6 of 11