INTERNATIONAL STANDARD ISO 10303-52:2011
TECHNICAL CORRIGENDUM 1

Published 2014-07-01

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

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

Part 52:

Integrated generic resource: Mesh based topology

TECHNICAL CORRIGENDUM 1

Systèmes d' automatisation industrielle et intégration – Représentation et échange de données de produits - Partie 52 Ressources génériques intégrées: Topologie du réseau

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to International Standard ISO 10303-52:2011 was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data.*

Included Bugzilla reports: Bug 3439, Bug 4617, 4813

ICS 25.040.40 **Ref. No. ISO 10303-52:2011/Cor.1:2014(E)**

Introduction

This Technical Corrigendum applies to ISO 10303-52:2011.

The purpose of the modifications to the text of ISO 10303-52:2001 is to correct errors relating to incorrect inclusion of a polygon cell shape. Corrections are made to the text, a table, to correct the EXPRESS in 3 entity definitions and to one EXPRESS function, and to update the document identifiers in annex B.

Modifications to the text of ISO 10303-52:2011

Page vi, tables

Delete the entry for table 2 and replace with:

Page 14, 4.3.1 array_based_unstructured_mesh

Some attributes referenced in the EXPRESS are not properly defined, Remove the EXPRESS definition of this entity and replce with:

EXPRESS specification:

```
*)
 ENTITY array_based_unstructured_mesh
   SUBTYPE OF (unstructured_mesh);
   cells  : ARRAY [1 : SELF\unstructured_mesh.cell_count] OF vertex_defined_cell;
 WHERE
   wr1:  SELF\mesh.index_count = 1;
END_ENTITY;
(*
```

Page 26, 4.3.1 explicit_unstructured_mesh

An attribute referenced in the EXPRESS is not properly defined, Remove the EXPRESS definition of this entity and replace with:

EXPRESS specification:

```
*)
ENTITY explicit_unstructured_mesh
  SUBTYPE OF (unstructured_mesh);
  explicit_model : fea_model;
  cells      : ARRAY [1 : SELF\unstructured_mesh.cell_count] OF UNIQUE element_represent
END_ENTITY;
(*
```

Page 27, table 2

This table contains information on polygon cells which are not included in ISO 10303-52. Remove table 2 completely and replace with:

Page 34, 4.3.21 vertex_defined_cell *The description of the* **vertices** *attribute contains an inappropriate reference to plygon. Remove the current description and replace with:*

Table 2 – Edges of triangle and quadrilateral cells

triangle		quadrilateral	
edge	**vertices**	**edge**	**vertices**
1	1, 2	1	1, 2
2	2, 3	2	2, 3
3	3, 1	3	3, 4
		4	4, 1

vertices: the vertices at the ends of cell edges, and within cell edges, cell faces and the interior of the cell. The position of a vertex or an edge node in the array depends on the shape of the cell as established graphically in Figures 10 through 39 in ISO 10303-104.

Page 36, 4.4.2 cell_counts *This function contains a case of polygon which does not exist. Further qualification has been added to other cases to avoid interaction with definitions in other schemas. Remove the EXPRESS definition of this function and replace with:*

EXPRESS specification:

```
*)
FUNCTION cell_counts(arg : vertex_defined_cell) : ARRAY[1:3] OF INTEGER;
LOCAL
  om1    : INTEGER := 0;       -- (order - 1)
  om1sq  : INTEGER := om1**2; -- (order - 1) squared
  vts    : INTEGER;            -- number of bounding vertices
  eds    : INTEGER;            -- number of edges
  qf     : INTEGER := 0;       -- number of quadrilateral faces
  tf     : INTEGER := 0;       -- number of triangular faces
  result : ARRAY [1:3] OF INTEGER := [0,0,0];
END_LOCAL;
  CASE arg.order OF
    linear    : om1 := 0;
    quadratic : om1 := 1;
    cubic     : om1 := 2;
    OTHERWISE : RETURN(result);
  END_CASE;
  om1sq := om1**2;
  CASE arg.shape OF
    cell_shape_0D.single :
        BEGIN
          vts := 1; eds := 0; qf := 0; tf := 0;
          result[1] := vts;
          result[2] := om1*eds;                    -- 0, 0, 0
          result[3] := 0;                          -- 0, 0, 0
        END;
    cell_shape_1D.line :
        BEGIN
```

```
          vts := 2; eds := 1; qf := 0; tf := 0;
          result[1] := vts;
          result[2] := om1*eds;                   -- 0, 1, 2
          result[3] := 0;                         -- 0, 0, 0
        END;
    cell_shape_2D.quadrilateral :
        BEGIN
          vts := 4; eds := 4; qf := 1; tf := 0;
          result[1] := vts;
          result[2] := om1*eds;                   -- 0, 4, 8
          result[3] := om1sq*qf;                  -- 0, 1, 4
        END;
    cell_shape_2D.triangle :
        BEGIN
          vts := 3; eds := 3; qf := 0; tf := 1;
          result[1] := vts;
          result[2] := om1*eds;                   -- 0, 3, 6
          result[3] := (om1-1)*tf;                --    0, 1
          CASE arg.order OF
            linear : result[3] := 0;              -- 0
          END_CASE;
        END;
    cell_shape_3D.hexahedron :
        BEGIN
          vts := 8; eds := 12; qf := 6; tf := 0;
          result[1] := vts;
          result[2] := om1*eds;                   -- 0, 12, 24
          result[3] := om1sq*(qf+om1);            -- 0, 7, 32
        END;
    cell_shape_3D.wedge :
        BEGIN
          vts := 6; eds := 9; qf := 3; tf := 2;
          result[1] := vts;
          result[2] := om1*eds;                   -- 0, 9,  18
          result[3] := om1sq*qf + om1*tf;         -- 0, 3, 16
        END;
    cell_shape_3D.tetrahedron :
        BEGIN
          vts := 4; eds := 6; qf := 0; tf := 4;
          result[1] := vts;
          result[2] := om1*eds;                   -- 0, 6, 12
          result[3] := (om1-1)*tf;                --    0, 4
          CASE arg.order OF
            linear : result[3] := 0;              -- 0
          END_CASE;
        END;
    cell_shape_3D.pyramid :
        BEGIN
          vts := 5; eds := 8; qf := 1; tf := 4;
          result[1] := vts;
          result[2] := om1*eds;                   -- 0, 8, 16
          result[3] := om1sq*qf + (om1-1)*tf;     --    1, 9
```