INTERNATIONAL STANDARD ISO 10303-41:1994
TECHNICAL CORRIGENDUM 1

Published 1999-06-15

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

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

Part 41:
Integrated generic resources: Fundamentals of product description and support

TECHNICAL CORRIGENDUM 1

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

Partie 41: Ressources génériques intégrées: Principes de description et de support de produits

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to International Standard ISO 10303-41:1994 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

Introduction

This document corrects ISO 10303-41:1994, Product data representation and exchange — Part 41: Integrated generic resources: Fundamentals of product description and support. The corrected document supersedes ISO 10303-41:1994.

The purpose of the modifications to the text of ISO 10303-41:1994 is to correct errors in the EXPRESS definitions likely to cause compilation problems, to replace guidance contained in the appendix that was incorrect for the usage of EXPRESS, to clarify the usage of the date entity, to replace the annex for the computer-interpretable EXPRESS with a URL reference, and to replace the object identifier for the document and the applicable schema.

ICS 25.040.40 **Ref. No. ISO 10303-41:1994/Cor.1:1999(E)**

Modifications to the text of ISO 10303-41:1994

Clause 2.3.5.1, p. 21

The EXPRESS specification for the FUNCTION acyclic_product_definition_formation_relationship does not find the acyclic relationships correctly. Remove the EXPRESS specification and replace with the following:

<u>EXPRESS specification</u>:

```
*)
FUNCTION acyclic_product_definition_formation_relationship
  (relation          : product_definition_formation_relationship;
   relatives         : SET [1:?] OF product_definition_formation;
   specific_relation : STRING) : LOGICAL;

  LOCAL
    x                : SET OF product_definition_formation_relationship;
  END_LOCAL;

  IF relation.relating_product_definition_formation IN
            relatives THEN
    RETURN (FALSE);
  END_IF;             -- IN is based in instance equality

  x := QUERY (pdf <* bag_to_set (USEDIN
       (relation.relating_product_definition_formation,
        'PRODUCT_DEFINITION_SCHEMA.' +
        'PRODUCT_DEFINITION_FORMATION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION_FORMATION')) |
         specific_relation IN TYPEOF (pdf));

  REPEAT I := 1 TO HIINDEX(x);             -- pre-checked loop
    IF NOT acyclic_product_definition_formation_relationship
      (x[i],
       relatives + relation.relating_product_definition_formation,
       specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;

  RETURN(TRUE);
END_FUNCTION;  -- acyclic_product_definition_formation_relationship
```

```
(*
```

Remove the Argument definitions and replace with the following:

Argument definitions:

relation: (input) the candidate **product_definition_formation_relationship** to be checked.

relatives: (input) the set of **product_definition_formation**s which the function is searching for in the **relating_product_definition_formation** parameter of the **relation** argument.

> NOTE 3 - When calling acyclic_product_definition_formation_relationship, the correct syntax to reference the **relatives** argument uses the aggregate initializer (e.g., ‘[<entity_name>.relating_xxx]’).

specific_relation: (input) the fully qualified entity name of a type **product_definition_formation_relationship** entity.

Clause 2.3.5.2, p. 23

The EXPRESS specification for the FUNCTION acyclic_product_definition_relationship does not find the acyclic relationships correctly. Remove the EXPRESS specification and replace with the following:

EXPRESS specification:

```
*)
FUNCTION acyclic_product_definition_relationship
  (relation          : product_definition_relationship;
   relatives         : SET [1:?] OF product_definition;
   specific_relation : STRING) : LOGICAL;

  LOCAL
    x                : SET OF product_definition_relationship;
  END_LOCAL;

  IF relation.relating_product_definition IN relatives THEN
    RETURN (FALSE);
  END_IF;             -- IN is based in instance equality

  x := QUERY (pd <* bag_to_set (USEDIN
       (relation.relating_product_definition,
        'PRODUCT_DEFINITION_SCHEMA.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION')) |
         specific_relation IN TYPEOF (pd));

  REPEAT I := 1 TO HIINDEX(x);             -- pre-checked loop
    IF NOT acyclic_product_definition_relationship
           (x[i],
            relatives + relation.relating_product_definition,
            specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;

  RETURN(TRUE);
END_FUNCTION;  -- acyclic_product_definition_relationship
(*
```

Remove the Argument definitions and replace with the following:

Argument definitions:

relation: (input) the candidate **product_definition_relationship** to be checked.

relatives: (input) the set of product_definitions which the function is searching for in the **relating_product_definition** parameter of the **relation** argument.

NOTE 3 - When calling acyclic_product_definition_relationship, the correct syntax to reference the **relatives** argument uses the aggregate initializer (e.g., ‘[<entity_name>.relating_xxx]’).

specific_relation: (input) the fully qualified entity name of a type **product_definition_relationship** entity.

Clause 2.4.4.3, p. 29
The EXPRESS specification for product_definition_shape contained an incorrect reference in WR1. Remove the EXPRESS specification and replace with the following:

EXPRESS specifcation:

```
*)
ENTITY product_definition_shape
 SUBTYPE OF (property_definition);
UNIQUE
 UR1:  SELF\property_definition.definition;
```

```
WHERE
 WR1:  NOT (‘PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_DEFINITION’
       IN TYPEOF (SELF\property_definition.definition));
END_ENTITY;
(*
```

Remove the following Formal proposition:

WR1: the **definition** attribute shall be a **characterized_product_definition**.

Replace the removed Formal proposition with the following:

WR1: the **definition** attribute shall be a **shape_definition.**

Clause 2.4.5, p. 32

The EXPRESS specification for the FUNCTION acyclic_shape_aspect_relationship does not find the acyclic relationships correctly. Remove the EXPRESS specification and replace with the following:

EXPRESS specification:

```
*)
FUNCTION acyclic_shape_aspect_relationship
  (relation          : shape_aspect_relationship;
   relatives         : SET [1:?] OF shape_aspect;
   specific_relation : STRING) : LOGICAL;

  LOCAL
    x                : SET OF shape_aspect_relationship;
  END_LOCAL;

  IF relation.relating_shape_aspect IN relatives THEN
    RETURN (FALSE);
  END_IF;             -- IN is based in instance equality

  x := QUERY (sa <* bag_to_set (USEDIN
       (relation.relating_shape_aspect,
        'PRODUCT_PROPERTY_DEFINITION_SCHEMA.' +
        'SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT')) |
         specific_relation IN TYPEOF (sa));

  REPEAT I := 1 TO HIINDEX(x);             -- pre-checked loop
```

```
    IF NOT acyclic_shape_aspect_relationship
           (x[i],
            relatives + relation.relating_shape_aspect,
            specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;

  RETURN(TRUE);
END_FUNCTION; -- acyclic_shape_aspect_relationship
(*
```

Remove the Argument definitions and replace with the following:

Argument definitions:

relation: (input) the candidate **shape_aspect_relationship** to be checked.

relatives: (input) the set of **shape_aspect**s which the function is searching for in the **relating_shape_aspect** parameter of the **relation** argument.

> NOTE 3 - When calling acyclic_shape_aspect_relationship, the correct syntax to reference the **relatives** argument uses the aggregate initializer (e.g., '[<entity_name>.relating_xxx]').

specific_relation: (input) the fully qualified entity name of a type **shape_aspect_relationship** entity.

Clause 2.5.4.2, p. 39

The EXPRESS specification for the FUNCTION relatives_of_shape_representations does not find the relatives correctly. Remove the EXPRESS specification and replace with the following:

EXPRESS specification:

```
*)
FUNCTION relatives_of_shape_representations
   (shape_representation_set : SET OF shape_representation) :
    SET OF shape_representation;

   FUNCTION local_relatives_of_shape_representations
      (shape_representation_set : SET OF shape_representation;
       total_reps : SET OF shape_representation) : SET OF
       shape_representation;
       LOCAL
         i               : INTEGER;
         local_shape_rep : SET OF shape_representation := [];
         local_srr       : SET OF shape_representation_relationship := [];
         local_total     : SET OF shape_representation := [];
       END_LOCAL;

       REPEAT i := 1 TO HIINDEX(shape_representation_set);
         local_srr := local_srr + QUERY (rr <* bag_to_set
           (USEDIN(shape_representation_set[i],
            'REPRESENTATION_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1'))|

'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_REPRESENTATION_RELATIONSHIP'
         IN TYPEOF (rr));
       END_REPEAT;

       REPEAT i := 1 TO HIINDEX(local_srr);
         IF 'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.'+
            'SHAPE_REPRESENTATION_RELATIONSHIP' IN TYPEOF(local_srr[i])
         THEN
           local_shape_rep := local_shape_rep + local_srr[i].rep_2;
         END_IF;
       END_REPEAT;
       IF SIZEOF (local_shape_rep - total_reps) = 0 THEN
          RETURN (shape_representation_set);
       ELSE
          local_total := total_reps + local_shape_rep;
       RETURN(local_shape_rep + (local_relatives_of_shape_representations
              (local_shape_rep - total_reps,
               local_total)));
       END_IF;
   END_FUNCTION;
   RETURN (local_relatives_of_shape_representations
   (shape_representation_set, shape_representation_set));
END_FUNCTION; -- relatives_of_shape_representations
(*
```

Clause 4.2.4, p. 56

The EXPRESS specification for the FUNCTION acyclic_document_relationship does not find the acyclic relationships correctly. Remove the EXPRESS specification and replace with the following:

<u>EXPRESS specification</u>:

```
*)
FUNCTION acyclic_document_relationship
  (relation          : document_relationship;
   relatives         : SET [1:?] OF document;
   specific_relation : STRING) : LOGICAL;

  LOCAL
    x                : SET OF document_relationship;
  END_LOCAL;

  IF relation.relating_document IN relatives THEN
    RETURN (FALSE);
  END_IF;             -- IN is based in instance equality

  x := QUERY (doc <* bag_to_set (USEDIN
       (relation.relating_document,
        'DOCUMENT_SCHEMA.' +
        'DOCUMENT_RELATIONSHIP.' +
        'RELATED_DOCUMENT'))  |
         specific_relation IN TYPEOF (doc));

  REPEAT I := 1 TO HIINDEX(x);             -- pre-checked loop
    IF NOT acyclic_document_relationship
           (x[i],
            relatives + relation.relating_document,
            specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;

  RETURN(TRUE);
END_FUNCTION;  -- acyclic_document_relationship
(*
```

Remove the Argument definitions and replace with the following:

Argument definitions:

relation: (input) the candidate **document_relationship** to be checked.

relatives: (input) the set of **document**s which the function is searching for in the **relating_document** parameter of the **relation** argument.

NOTE 3 - When calling acyclic_document, the correct syntax to reference the **relatives** argument uses the aggregate initializer (e.g., ‘[<entity_name>.relating_xxx]’).

specific_relation: (input) the fully qualified entity name of a type **document_relationship** entity.

Clause 4.3.5.1, p. 64

The EXPRESS specification for the FUNCTION acyclic_action_relationship does not find the acyclic relationships correctly. Remove the EXPRESS specification and replace with the following:

EXPRESS specification:

```
*)
FUNCTION acyclic_action_relationship
  (relation          : action_relationship;
   relatives         : SET [1:?] OF action;
   specific_relation : STRING) : LOGICAL;

  LOCAL
    x                : SET OF action_relationship;
  END_LOCAL;

  IF relation.relating_action IN relatives THEN
    RETURN (FALSE);
  END_IF;              -- IN is based in instance equality

  x := QUERY (actn <* bag_to_set (USEDIN
       (relation.relating_action,
        'ACTION_SCHEMA.' +
        'ACTION_RELATIONSHIP.' +
        'RELATED_ACTION')) |
         specific_relation IN TYPEOF (actn));

  REPEAT I := 1 TO HIINDEX(x);             -- pre-checked loop
    IF NOT acyclic_action_relationship
           (x[i],
            relatives + relation.relating_action,
            specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;

  RETURN(TRUE);
```

```
END_FUNCTION; -- acyclic_action_relationship
(*
```

Remove the Argument definitions and replace with the following:

Argument definitions:

relation: (input) the candidate **action_relationship** to be checked.

relatives: (input) the set of **action**s which the function is searching for in the **relating_action** parameter of the **relation** argument.

> NOTE 3 - When calling acyclic_action_relationship, the correct syntax to reference the **relatives** argument uses the aggregate initializer (e.g., '[<entity_name>.relating_xxx]').

specific_relation: (input) the fully qualified entity name of a type **action_relationship** entity.

Clause 4.3.5.2, p. 65
The EXPRESS specification for the FUNCTION acyclic_action_resource_relationship does not find the acyclic relationships correctly. Remove the EXPRESS specification and replace with the following:

EXPRESS specification:

```
*)
FUNCTION acyclic_action_resource_relationship
  (relation          : action_resource_relationship;
   relatives         : SET [1:?] OF action_resource;
   specific_relation : STRING) : LOGICAL;

  LOCAL
    x                : SET OF action_resource_relationship;
  END_LOCAL;

  IF relation.relating_resource IN relatives THEN
    RETURN (FALSE);
  END_IF;             -- IN is based in instance equality

  x := QUERY (ar <* bag_to_set (USEDIN
       (relation.relating_resource,
        'ACTION_SCHEMA.' +
        'ACTION_RESOURCE_RELATIONSHIP.' +
        'RELATED_RESOURCE')) |
         specific_relation IN TYPEOF (ar));

  REPEAT I := 1 TO HIINDEX(x);             -- pre-checked loop
    IF NOT acyclic_action_resource_relationship
           (x[i],
            relatives + relation.relating_resource,
            specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;

  RETURN(TRUE);
END_FUNCTION; -- acyclic_action_resource_relationship
(*
```

Remove the Argument definitions and replace with the following:

<u>Argument definitions</u>:

relation: (input) the candidate **action_resource_relationship** to be checked.

relatives: (input) the set of **action_resource**s which the function is searching for in the **relating_resource** parameter of the **relation** argument.

> NOTE 3 - When calling acyclic_action_resource_relationship, the correct syntax to reference the **relatives** argument uses the aggregate initializer (e.g., '[<entity_name>.relating_xxx]').

specific_relation: (input) the fully qualified entity name of a type **action_resource_relationship** entity.

Clause 4.3.5.3, p. 67

The EXPRESS specification for the FUNCTION acyclic_action_method_relationship does not find the acyclic relationships correctly. Remove the EXPRESS specification and replace with the following:

<u>EXPRESS specification</u>:

```
*)
FUNCTION acyclic_action_method_relationship
  (relation          : action_method_relationship;
   relatives         : SET [1:?] OF action_method;
   specific_relation : STRING) : LOGICAL;
```