INTERNATIONAL STANDARD

ISO 10303-521

First edition
2003-03-01

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

Part 521:
Application interpreted construct: Manifold subsurface

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

Partie 521: Construction interprétée d'application: Surface complexe à topologie non eulérienne

Reference number
ISO 10303-521:2003(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

Contents

Page

Figures

Tables

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 10303-521 was prepared by Technical Committee ISO/TC 184/SC 4. *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data.*

This International Standard is organised as a series of parts, each published separately. The structure of this International Standard is decribed in ISO 10303-1.

Each part of this International Standard is a member of one of the following series: decription methods, implementation methods, conformance testing methodology and framework, integrated generic resources, integrated application resources, application protocols, abstract test suites, application interpreted constructs, and application modules. This part is a member of the application interpreted construct series.

A complete list of parts of ISO 10303 is available from Internet:

`<http://www.tc184-sc4.org/titles/STEP_Titles.rtf>`

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and as a basis for archiving.

This part of ISO 10303 is a member of the application interpreted construct series. An application interpreted construct (AIC) provides a logical grouping of interpreted constructs that supports a specific functionality for the usage of product data across multiple application contexts. An interpreted construct is a common interpretation of the integrated resources that supports shared information requirements among application protocols.

This document specifies the application interpreted construct for manifold subsurface. This provides the definition of a shape representation containing open shells, each of which is identified as being part of another open or closed shell. The shells are defined using faces with explicit topology and fully defined geometry. The faces of the shells defined in this AIC either use the **advanced_face** definition from ISO 10303-511, or have similar properties.

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

Part 521:
Application interpreted construct: Manifold subsurface

1 Scope

This part of ISO 10303 specifies the interpretation of the integrated resources to satisfy the requirement for the definition of a shape representation containing open shells defined as connected face sub sets. The domain of each connected face sub set is part of the domain of another open, or closed shell. Within the connected face sub set individual faces or edges may be identified as being subfaces, or subedges.

The following are within the scope of this part of ISO 10303:

— 3D geometry;

— advanced faces;

— connected face sub sets;

— mappings and geometric transformations;

— open shells;

— relationships between domains of topological objects;

— subedges;

— subfaces;

— unbounded geometry with associated topological boundaries;

— use of topology to bound geometric entities.

The following are outside the scope of this part of ISO 10303:

— 2D geometry other than for the definition of a pcurve in the parameter space of a surface;

— boundary representation solid models;

— bounded curves other than polylines and B-spline curves;

— bounded surfaces other than B-spline surfaces;

— non-manifold geometry;

— offset curves and surfaces;

— unbounded geometry without topological boundaries.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8824-1: 1998, *Information technology - Abstract syntax notation one (ASN.1) Part 1: Specification of basic notation*

ISO 10303-1: 1994, *Industrial automation systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles*

ISO 10303-11: 1994, *Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual*

ISO 10303-41: 2000, *Industrial automation systems and integration - Product data representation and exchange - Part 41: Integrated generic resource: Fundamentals of product description and support*

ISO 10303-42: 2000, *Industrial automation systems and integration - Product data representation and exchange - Part 42: Integrated generic resource: Geometric and topological representation*

ISO 10303-43: 2000, *Industrial automation systems and integration - Product data representation and exchange - Part 43: Integrated generic resource: Representation structures*

ISO 10303-202: 1995, *Industrial automation systems and integration - Product data representation and exchange - Part 202: Application protocol: Associative draughting*

ISO 10303-511: 2001, *Industrial automation systems and integration - Product data representation and exchange - Part 511: Application interpreted construct: Topologically bounded surface*

3 Terms, definitions and abbreviations

3.1 Terms defined in ISO 10303-1

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-1 apply.

— application;

— application context;

— application protocol;

— implementation method;

— integrated resource;

— interpretation;

— product data.

3.2 Terms defined in ISO 10303-42

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-42 apply.

— arcwise connected;

— axi-symmetric;

— boundary;

— bounds;

— coordinate space;

— curve;

— domain;

— extent;

— open curve;

— orientable;

— surface;

— topological sense.

3.3 Terms defined in ISO 10303-202

For the purposes of this part of ISO 10303, the following term defined in ISO 10303-202 applies.

— application interpreted construct.

3.4 Terms defined in ISO 10303-511

For the purposes of this part of ISO 10303, the following term defined in ISO 10303-511 applies.

— advanced face.

3.5 Other definitions

3.5.1
Connected face subset
set of arcwise connected faces whose domain is a portion of the domain of an existing connected face set

NOTE 1 In this part of ISO 10303 a connected face subset is also required to be of type open shell.

3.5.2
manifold subsurface shape representation
shape representation containing connected face sub sets in the form of open shells

NOTE 1 Each open shell has a domain that is part of the domain of another connected face set. The faces of the connected face sub set may be defined as subfaces.

3.5.3
subedge
edge whose domain is a connected portion of the domain of another edge

3.5.4
subface
face whose domain is a connected portion of the domain of another face

NOTE 1 In this part of ISO 10303 a subface has a domain which is part of the domain of an advanced face.

3.6 Abbreviations

For the purposes of this part of ISO 10303, the following abbreviations apply.

AIC Application Interpreted Construct
AP Application Protocol

4 EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources and contains the types, entity specializations, and functions that are specific to this part of ISO 10303.

NOTE 1 There may be subtypes and items of select lists that appear in the integrated resources that are not imported into the AIC. Constructs are eliminated from the subtype tree or select list through the use of the implicit interface rules of ISO 10303-11. References to eliminated constructs are outside the scope of the AIC. In some cases, all items of the select list are eliminated. Because AICs are intended to be implemented in the context of an application protocol, the items of the select list will be defined by the scope of the application protocol.

This application interpreted construct provides a consistent set of geometric and topological entities for the definition of manifold subsurface shape representation. Faces may be advanced faces or subfaces referencing an advanced face. Edges are required to be subedges, or, to have their geometry defined by curves. The highest level entity in this AIC is **manifold_subsurface_shape_representation** which is a specialised type of **shape_representation** (see ISO 10303-41). The rules on this entity ensure that the topology and geometry are fully defined.

NOTE 2 This AIC uses all the entities and types from the topologically bounded surface AIC (**aic_topologically_bounded_surface**). ISO 10303-511 should be referred to in order to obtain the complete data set.

EXPRESS specification:

```
*)
SCHEMA aic_manifold_subsurface;
 USE FROM aic_topologically_bounded_surface; -- ISO 10303-511

 USE FROM geometry_schema    -- ISO  10303-42
     (cartesian_transformation_operator_3d);
 USE FROM topology_schema  -- ISO  10303-42
     (closed_shell,
      connected_face_set,
      connected_face_sub_set,
      face,
      open_shell,
      subedge,
      subface);

 USE FROM representation_schema(mapped_item);   -- ISO  10303-43

 USE FROM product_property_representation_schema   -- ISO  10303-41
     (shape_representation);
(*
```

NOTE 3 The schemas referenced above can be found in the following parts of ISO 10303:

geometry_schema	ISO 10303-42: 2000
topology_schema	ISO 10303-42: 2000
representation_schema	ISO 10303-43
product_property_representation_schema	ISO 10303-41
aic_topologically_bounded_surface	ISO 10303-511

4.1 Fundamental concepts and assumptions

The following entities[1)] are intended to be independently instantiated in the application protocol schemas that use this AIC:

— advanced_face [511];

— axis2_placement_2d [511];

— axis2_placement_3d [511];

— brep_with_voids;

— bezier_curve [511];

— bezier_surface [511];

— b_spline_curve_with_knots [511];

— b_spline_surface_with_knots [511];

— cartesian_point [511];

— cartesian_transformation_operator_3d;

— circle [511];

— closed_shell;

— conical_surface [511];

— definitional_representation [511];

— degenerate_toroidal_surface [511];

— connected_face_sub_set;

— cylindrical_surface [511];

[1)]The entities marked [511] are defined in the **aic_topologically_bounded_surface**

— direction [511];
— edge_curve [511];
— edge_loop [511];
— ellipse [511];
— face_bound [511];
— face_outer_bound [511];
— geometric_representation_context [511];
— hyperbola [511];
— line [511];
— manifold_subsurface_shape_representation;
— mapped_item;
— open_shell;
— parabola [511];
— parametric_representation_context [511];
— pcurve [511];
— plane [511];
— polyline [511];
— quasi_uniform_curve [511];
— quasi_uniform_surface [511];
— rational_b_spline_curve [511];
— rational_b_spline_surface [511];
— representation_map;
— spherical_surface [511];
— subedge;

— subface;

— surface_of_linear_extrusion [511];

— surface_of_revolution [511];

— toroidal_surface [511];

— uniform_curve [511];

— uniform_surface [511];

— vector [511];

— vertex_loop [511];

— vertex_point [511].

An application protocol that uses this AIC shall ensure that the **shape_representation** entity is instantiated as a **manifold_subsurface_shape_representation**.

4.2 aic_manifold_subsurface schema entity definition: manifold_subsurface_shape_representation

The **manifold_subsurface_shape_representation** is a type of **shape_representation** in which the shape of a product is represented by specialisations of **connected_face_sub_set** entities.

Each **connected_face_sub_set** is required to be also of type **open_shell**.

The **face**s of the **connected_face_sub_set**s are required to be of type **advanced_face**, or, to be of type **subface**.

EXPRESS specification:

```
*)
 ENTITY manifold_subsurface_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
   WR1: SIZEOF (QUERY (it <* SELF.items |
          NOT (SIZEOF (['AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET',
                'AIC_MANIFOLD_SUBSURFACE.MAPPED_ITEM',
                'AIC_MANIFOLD_SUBSURFACE.AXIS2_PLACEMENT_3D'] *
                   TYPEOF(it)) = 1))) = 0;
   WR2: SIZEOF (QUERY (it <* SELF.items |
         SIZEOF(['AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET',
         'AIC_MANIFOLD_SUBSURFACE.MAPPED_ITEM'] * TYPEOF(it)) =1 )) > 0;
```

```
WR3: SIZEOF (QUERY (mi <*  QUERY (it <* items |
              'AIC_MANIFOLD_SUBSURFACE.MAPPED_ITEM' IN TYPEOF(it)) |
NOT ('AIC_MANIFOLD_SUBSURFACE.MANIFOLD_SUBSURFACE_SHAPE_REPRESENTATION' IN
          TYPEOF(mi\mapped_item.mapping_source.
                        mapped_representation)))) = 0;
WR4: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
        'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET' IN TYPEOF(it)) |
       NOT('AIC_MANIFOLD_SUBSURFACE.OPEN_SHELL' IN TYPEOF(cfss)))) = 0;
WR5: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
        'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET' IN TYPEOF(it)) |
       NOT( (('AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET' IN
                TYPEOF(cfss.parent_face_set)) AND
         (SIZEOF (QUERY (fac <* cfss.parent_face_set.cfs_faces | NOT
              advanced_face_properties(fac))) = 0)) OR
          (SIZEOF (QUERY (fac <* cfss.parent_face_set.cfs_faces | NOT
           ('AIC_MANIFOLD_SUBSURFACE.ADVANCED_FACE' IN TYPEOF(fac)))) = 0)
                 ))) = 0;
WR6: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
        'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET' IN TYPEOF(it)) |
       ( SIZEOF (QUERY (fac <* cfss\connected_face_set.cfs_faces  | NOT
           advanced_face_properties(fac))) = 0))) = 0;
WR7: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
       'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET' IN TYPEOF(it)) |
          NOT (SIZEOF (QUERY(fcs <* cfss\connected_face_set.cfs_faces |
            ('AIC_MANIFOLD_SUBSURFACE.SUBFACE' IN TYPEOF(fcs)) AND
         NOT (SIZEOF(QUERY (elp_fbnds <* QUERY (bnds <* fcs.bounds |
      'AIC_MANIFOLD_SUBSURFACE.EDGE_LOOP' IN TYPEOF(bnds.bound)) |
             NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
               NOT(('AIC_MANIFOLD_SUBSURFACE.EDGE_CURVE' IN
         TYPEOF(oe.edge_element)) OR
        ('AIC_MANIFOLD_SUBSURFACE.SUBEDGE' IN
         TYPEOF(oe.edge_element)) ))) = 0
                ))) = 0
              ))) = 0
          ))) = 0;
WR8: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
        'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUBSET' IN TYPEOF(it)) |
          NOT (SIZEOF (QUERY(fcs <* cfss\connected_face_set.cfs_faces |
        ('AIC_MANIFOLD_SUBSURFACE.SUBFACE' IN TYPEOF(fcs)) AND
            NOT (SIZEOF(QUERY (elp_fbnds <* QUERY (bnds <* fcs.bounds |
      'AIC_MANIFOLD_SUBSURFACE.EDGE_LOOP' IN TYPEOF(bnds.bound)) |
              NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
       NOT(('AIC_MANIFOLD_SUBSURFACE.VERTEX_POINT' IN TYPEOF(oe.edge_start))
             AND ('AIC_MANIFOLD_SUBSURFACE.VERTEX_POINT' IN
              TYPEOF(oe.edge_end))
         ))) = 0
           ))) = 0
            ))) = 0
                ))) = 0;
WR9: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
        'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET' IN TYPEOF(it)) |
          NOT (SIZEOF (QUERY(fcs <* cfss\connected_face_set.cfs_faces |
```

```
            ( 'AIC_MANIFOLD_SUBSURFACE.SUBFACE' IN TYPEOF(fcs)) AND
            ( NOT (SIZEOF(QUERY  (bnds <* fcs.bounds |
       NOT (SIZEOF (['AIC_MANIFOLD_SUBSURFACE.EDGE_LOOP',
              'AIC_MANIFOLD_SUBSURFACE.VERTEX_LOOP'] *
                 TYPEOF(bnds.bound)) = 1 )
                 )) = 0)
               ))) = 0
           ))) = 0;
 WR10: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
         'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUB_SET' IN TYPEOF(it)) |
           NOT (SIZEOF (QUERY(fcs <* cfss\connected_face_set.cfs_faces |
            ( 'AIC_MANIFOLD_SUBSURFACE.SUBFACE' IN TYPEOF(fcs)) AND
            ( NOT (SIZEOF(QUERY (elp_fbnds <* QUERY (bnds <* fcs.bounds |
       'AIC_MANIFOLD_SUBSURFACE.EDGE_LOOP' IN TYPEOF(bnds.bound)) |
               NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
                 NOT (SIZEOF (['AIC_MANIFOLD_SUBSURFACE.LINE',
                       'AIC_MANIFOLD_SUBSURFACE.CONIC',
                       'AIC_MANIFOLD_SUBSURFACE.POLYLINE',
                       'AIC_MANIFOLD_SUBSURFACE.SURFACE_CURVE',
                       'AIC_MANIFOLD_SUBSURFACE.B_SPLINE_CURVE' ] *
          TYPEOF(oe.edge_element\edge_curve.edge_geometry)) = 1 )
             )) = 0
             ))) = 0
               )))) = 0
                  ))) = 0;
 WR11: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
         'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUBSET' IN TYPEOF(it)) |
           NOT (SIZEOF (QUERY(fcs <* cfss\connected_face_set.cfs_faces |
            ( 'AIC_MANIFOLD_SUBSURFACE.SUBFACE' IN TYPEOF(fcs)) AND
             (NOT (SIZEOF(QUERY (elp_fbnds <* QUERY (bnds <* fcs.bounds |
       'AIC_MANIFOLD_SUBSURFACE.EDGE_LOOP' IN TYPEOF(bnds.bound)) |
              NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
                 ('AIC_MANIFOLD_SUBSURFACE.SURFACE_CURVE' IN
          TYPEOF(oe.edge_element\edge_curve.edge_geometry)) AND
          (NOT ((SIZEOF (QUERY (sc_ag <*
          oe.edge_element\edge_curve.edge_geometry\
          surface_curve.associated_geometry |
          NOT ('AIC_TOPOLOGICALLY_BOUNDED_SURFACE.PCURVE' IN
          TYPEOF(sc_ag)))) = 0)))
             )) = 0
             ))) = 0
                 )))) = 0
                  ))) = 0;
 WR12: SIZEOF (QUERY (cfss <* QUERY (it <* SELF.items |
         'AIC_MANIFOLD_SUBSURFACE.CONNECTED_FACE_SUBSET' IN TYPEOF(it)) |
           NOT (SIZEOF (QUERY(fcs <* cfss\connected_face_set.cfs_faces |
            ( 'AIC_MANIFOLD_SUBSURFACE.SUBFACE' IN TYPEOF(fcs)) AND
             (NOT (SIZEOF(QUERY (elp_fbnds <* QUERY (bnds <* fcs.bounds |
       'AIC_MANIFOLD_SUBSURFACE.EDGE_LOOP' IN TYPEOF(bnds.bound)) |
              NOT (SIZEOF (QUERY (oe <* elp_fbnds.bound\path.edge_list |
                 ('AIC_MANIFOLD_SUBSURFACE.POLYLINE' IN
          TYPEOF(oe.edge_element\edge_curve.edge_geometry)) AND
```

```
            (NOT (SIZEOF (oe\oriented_edge.edge_element\
                  edge_curve.edge_geometry\polyline.points) >= 3))
               )) = 0
               ))) = 0
                   )))) = 0
                    ))) = 0;
 END_ENTITY;
(*
```

Formal propositions:

WR1: The **items** in a **manifold_subsurface_shape_representation** shall be **connected_face_sub_set**s, **mapped_item**s, or **axis2_placement_3d**s.

WR2: At least one of the **items** shall be either a **connected_face_sub_set** or a **mapped_item**.

WR3: For any **mapped_item**, the **mapped_representation** of its **mapping_source** shall be a **manifold_subsurface_shape_representation**.

WR4: Any instance of **connected_face_sub_set** in the **items** set shall be of type **connected_face_sub_set** AND **open_shell**.

WR5: For any instance of **connected_face_sub_set** the **face**s of the **parent_face_set** shall be of type **advanced_face** or the **parent_face_set** shall be an instance of **connected_face_sub_set** whose faces are related to **advanced_face**s.

NOTE 1 The **parent_face_set** may be one of the subtypes, **open_shell**, or **closed_shell** of **connected_face_set**.

WR6: For any instance of **connected_face_subset** each **face** of the **cfs_faces** attribute shall be of type **advanced_face** or of type **subface** and, directly or indirectly, reference an **advanced_face**.

NOTE 2 This property is verified by the **advanced_face_properties** function.

WR7: For any instance of **subface** in the **cfs_faces** of a **connected_face_sub_set** instance the bounding **edge**s shall be of type **subedge** or of type **edge_curve**.

WR8: For any instance of **subface** in the **cfs_faces** of a **connected_face_sub_set** instance all vertices used in the **face** definition shall be of type **vertex_point**.

WR9: The bounds of any **subface** in the **cfs_faces** of a **connected_face_sub_set** instance shall be of type **edge_loop**, or, of type **vertex_loop**.

WR10: The types of **curve** used to define the geometry of an **edge_curve** used in the definition of a **subface** shall be restricted to **line**s, **conic**s, **polyline**s, **surface_curve**s or **b_spline_curve**s.

WR11: If a **surface_curve** is used as part of a face bound of a **subface** the **associated_geometry** attribute shall reference a **pcurve**.

WR12: If a **polyline** is used as part of a face bound of a **subface** it shall contain at least 3 points.

NOTE 3 Rules WR8 to WR13 ensure that the definition of the bounds of a **subface** are consistent with those of an **advanced_face**.

4.3 aic_manifold_subsurface schema function definition: advanced_face_properties

The **advanced_face_properties** function checks the properties of a **face** to determine if it is of type **advanced_face**, or, if it is a **subface** directly, or indirectly, referencing an **advanced_face** as **parent_face**. This check is carried out recursively and a TRUE result is returned if the face either is an **advanced_face**, or, via the **parent_face** attribute of a **subface** references an **advanced_face**. In all other cases a FALSE result is returned.

EXPRESS specification:

```
*)
FUNCTION advanced_face_properties (testface : face) : BOOLEAN;
(* return TRUE if testface is of type advanced_face *)
IF 'AIC_MANIFOLD_SUBSURFACE.ADVANCED_FACE' IN TYPEOF(testface) THEN
    RETURN (TRUE);
END_IF;
(* if testface is a subface recursively test the parent_face,
return FALSE for all other types of face *)
IF  ('AIC_MANIFOLD_SUBSURFACE.SUBFACE' IN TYPEOF(testface)) THEN
   RETURN(advanced_face_properties(testface.parent_face));
   ELSE  RETURN (FALSE);
END_IF;
END_FUNCTION;
(*
```

Argument definitions:

testface: (input) The **face** which is to be tested for **advanced_face** properties.

result: (output) A BOOLEAN variable which is TRUE if **testface** is of type **advanced_face**. or is of type **subface** and references an **advanced_face**.

EXPRESS specification:

```
 *)
   END_SCHEMA; -- end AIC_MANIFOLD_SUBSURFACE SCHEMA
 (*
```

Annex A
(normative)

Short names of entities

Table A.1 provides the short names of entities specified in the EXPRESS listing of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table A.1 – Short names of entities

Entity name	Short name
MANIFOLD_SUBSURFACE_SHAPE_REPRESENTATION	MSSO

Annex B
(normative)

Information object registration

B.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(521) version(1) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2 Schema identification

To provide for unambiguous identification of the aic_manifold_subsurface in an open information system, the object identifier

{ iso standard 10303 part(521) version(1) schema(1) aic-manifold-subsurface(1) }

is assigned to the aic_manifold_subsurface schema (see 4). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex C
(informative)

Computer-interpretable listings

This annex provides a listing of the EXPRESS entity names and corresponding short names as specified in this Part of ISO 10303 without comments or other explanatory text. This annex is available in computer-interpretable form and can be found at the following URLs:

Short names: <http://www.tc184-sc4.org/Short_Names/>

EXPRESS: <http://www.tc184-sc4.org/EXPRESS/>

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@tc184-sc4.org.

NOTE – The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex D
(informative)

EXPRESS-G diagrams

Figures D.1 through D.9 correspond to the EXPRESS generated from the short listing given in clause 4 using the interface specifications of ISO 10303-11. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

NOTE 1 The following select types are interfaced into the AIC expanded listing according to the implicit interface rules of ISO 10303-11. These select types are not used by other entities in this part of ISO 10303.

— geometric_set_select;

— trimming_select;

— vector_or_direction.

NOTE 2 The rules on **advanced_face** and on **manifold_subsurface_shape_representation** exclude the instantiation of some entities which are implicitly interfaced and, therefore, shown in the diagrams. These entities are marked with a ∗ in the diagrams.

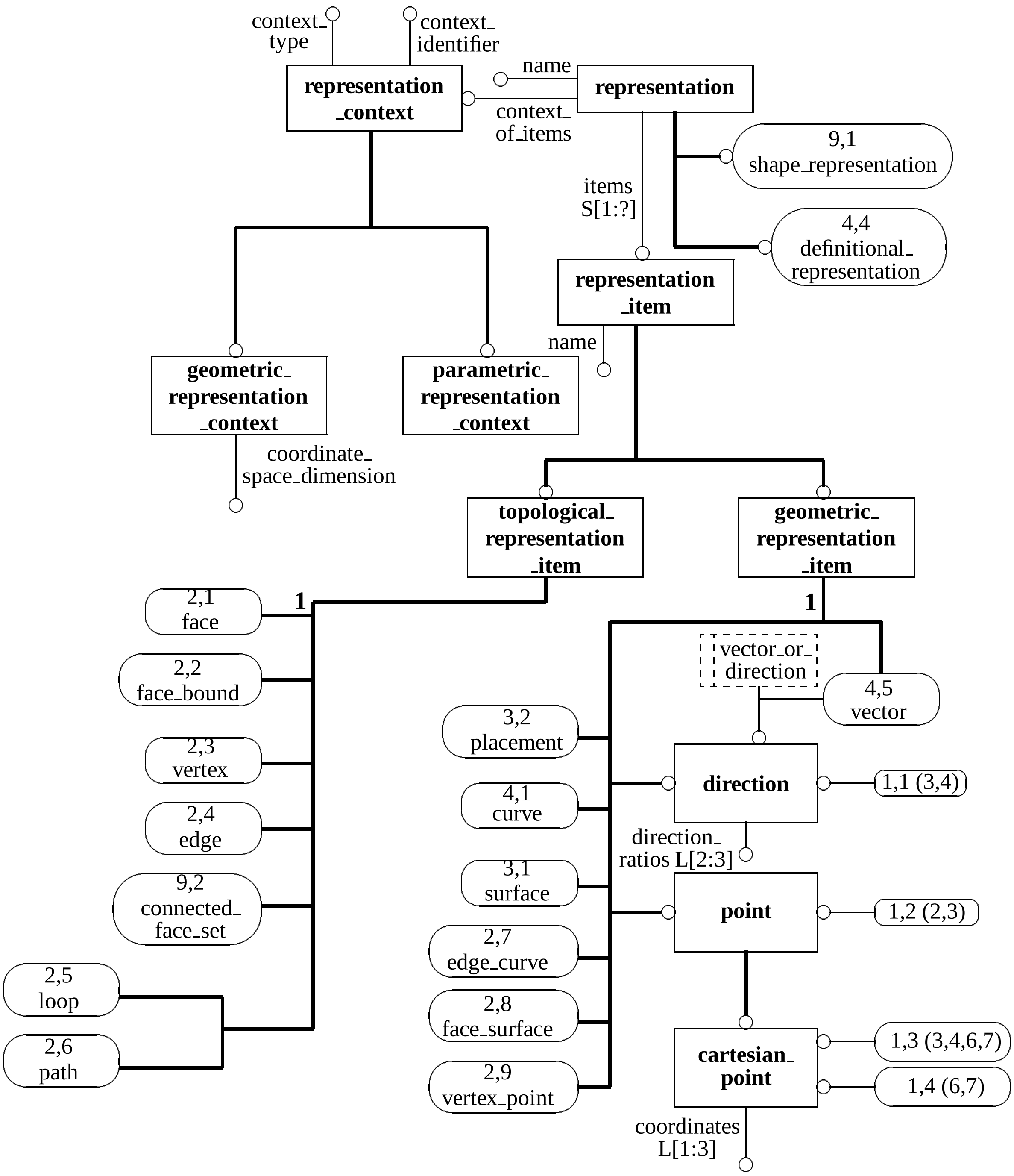


Figure D.1 – aic_manifold_subsurface EXPRESS-G diagram page 1 of 9

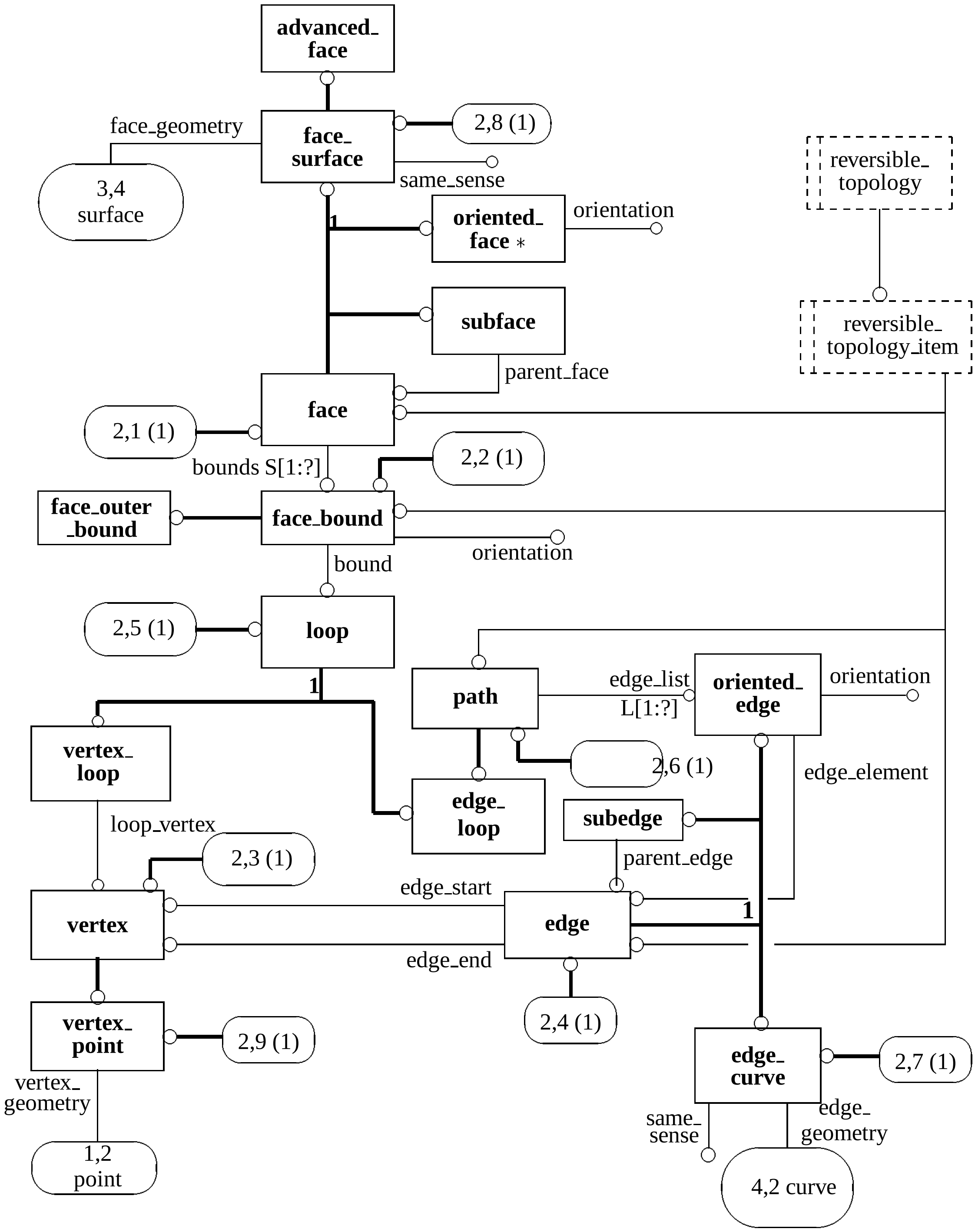


Figure D.2 – aic_manifold_subsurface EXPRESS-G diagram page 2 of 9

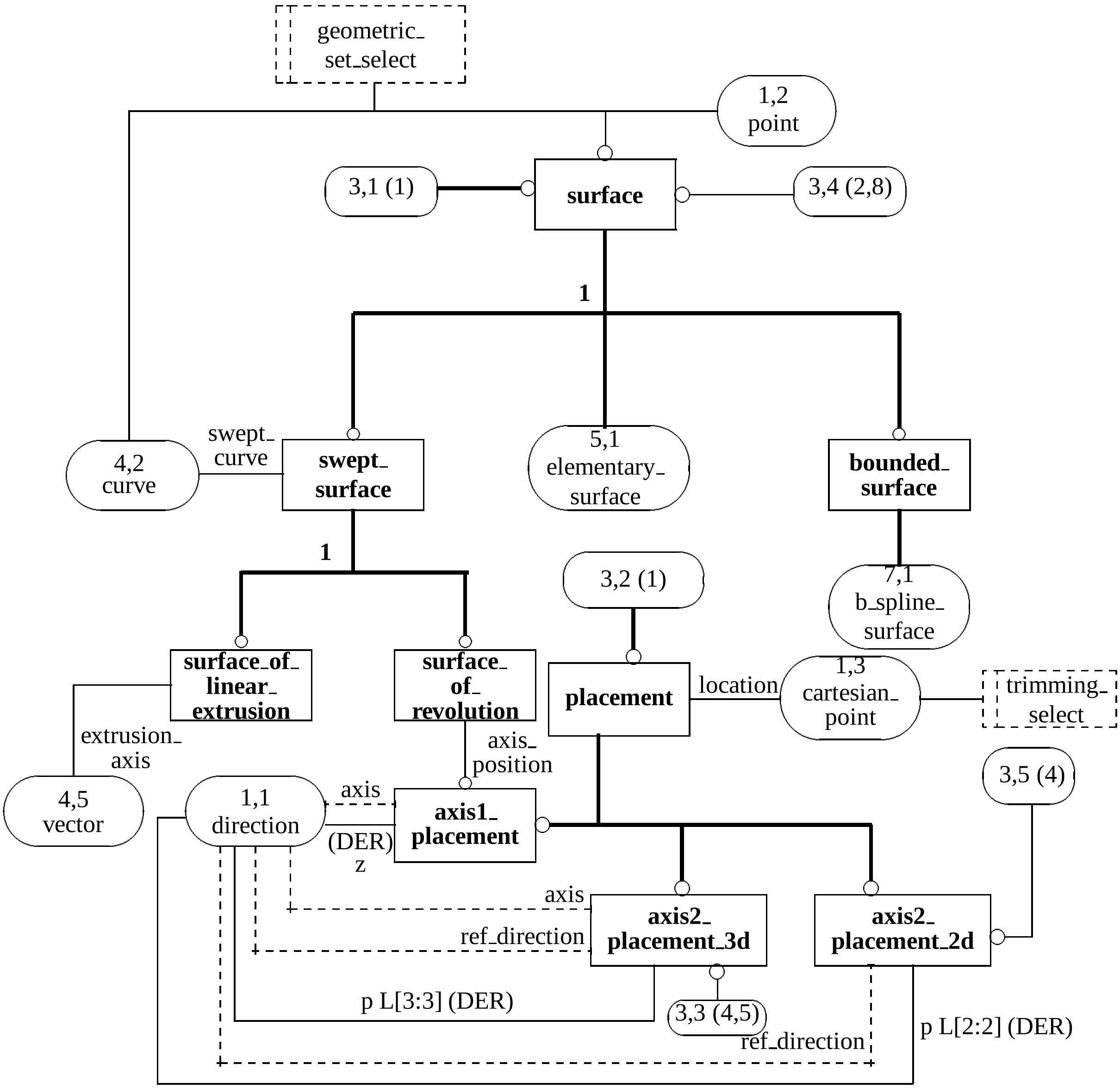


Figure D.3 – aic_manifold_subsurface EXPRESS-G diagram page 3 of 9

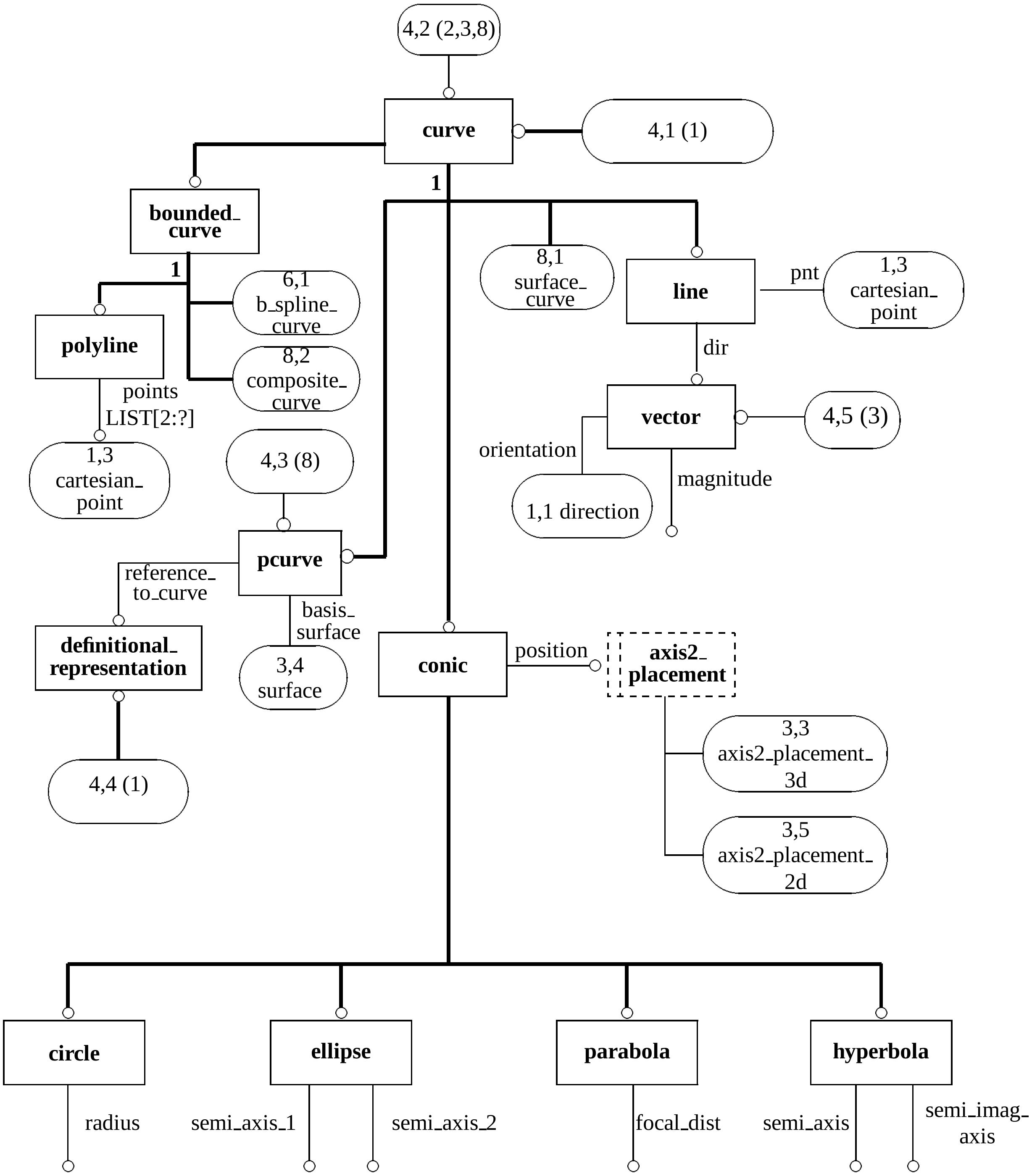


Figure D.4 – aic_manifold_subsurface EXPRESS-G diagram page 4 of 9

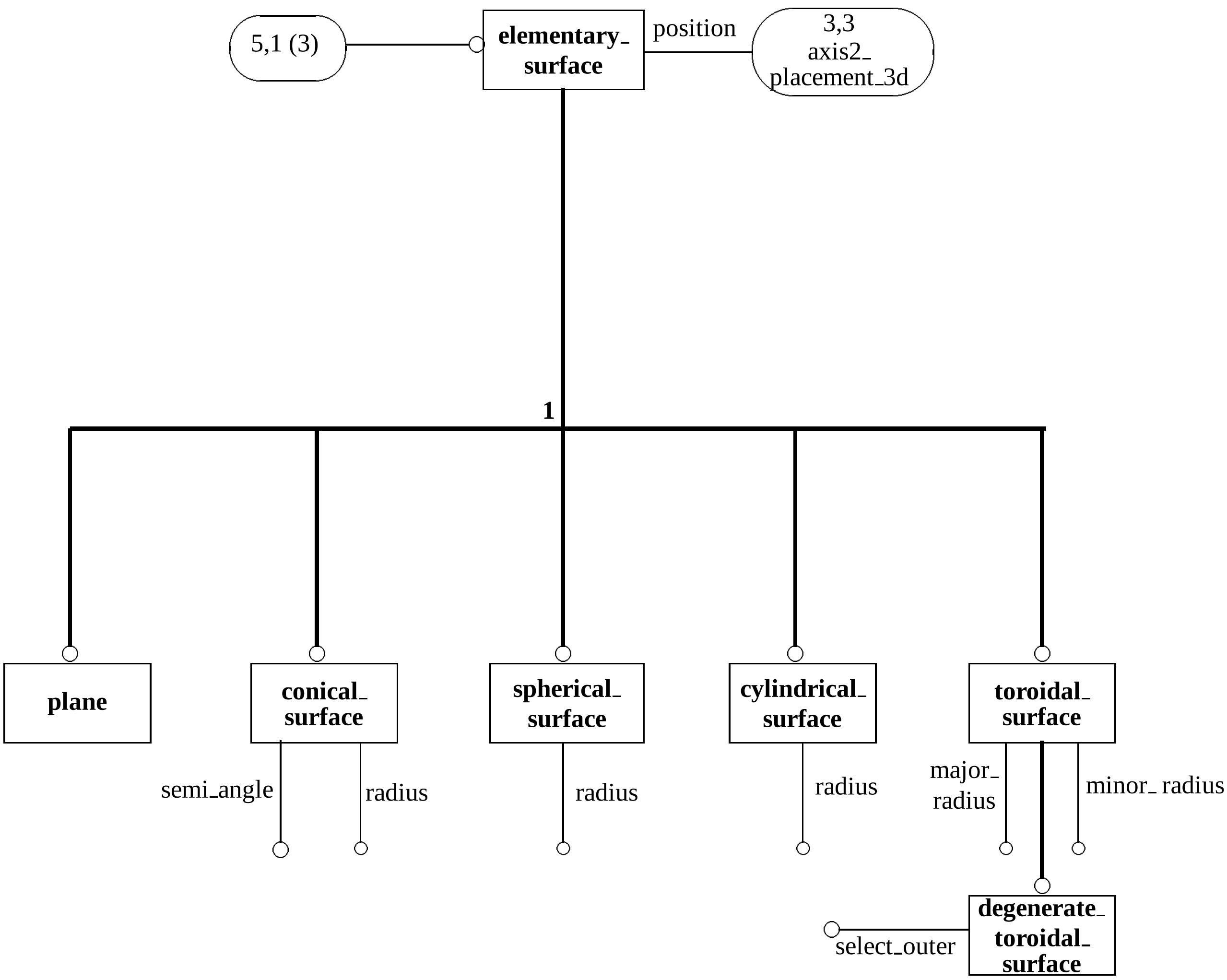


Figure D.5 – aic_manifold_subsurface EXPRESS-G diagram page 5 of 9

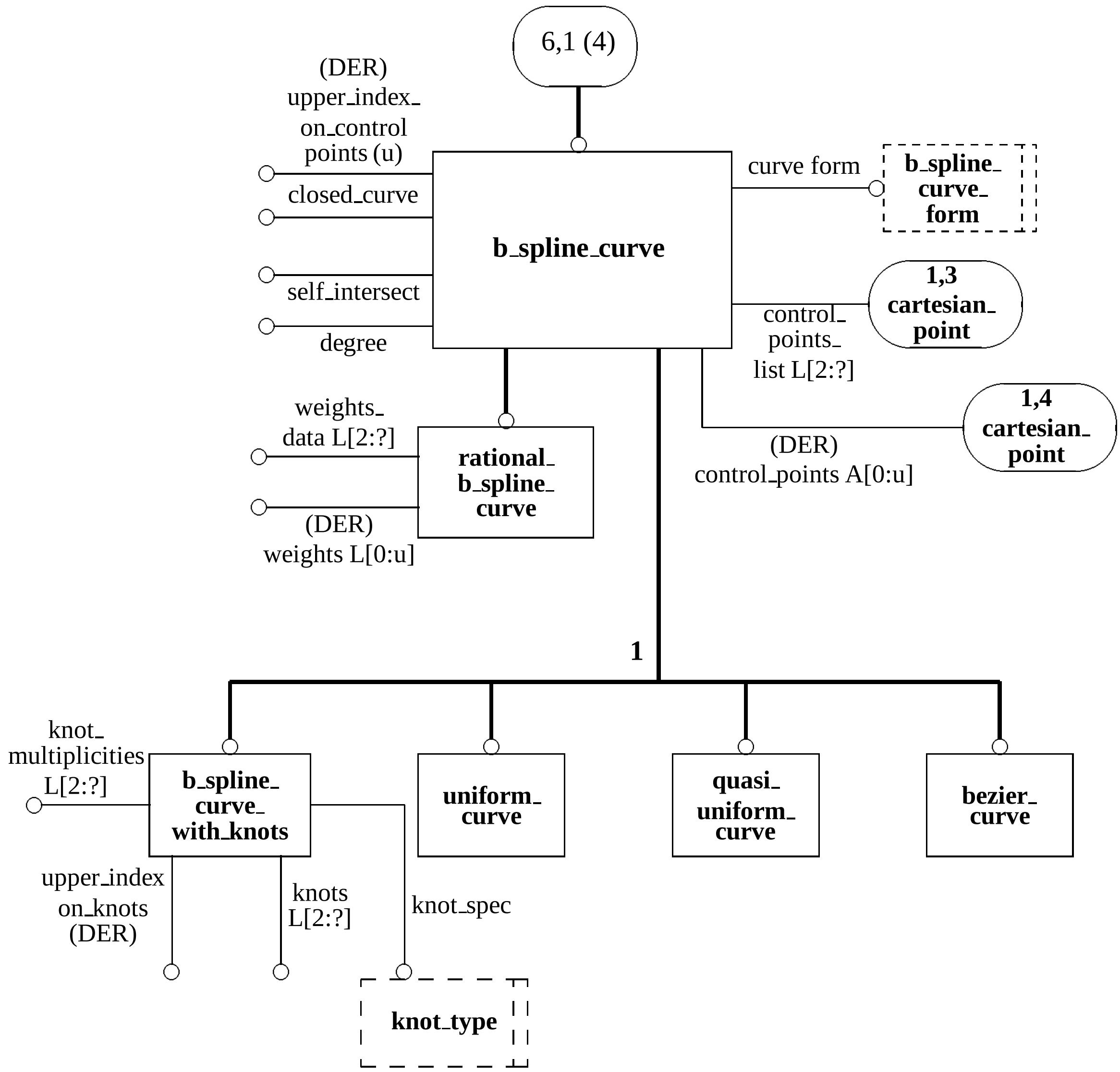


Figure D.6 – aic_manifold_subsurface EXPRESS-G diagram page 6 of 9

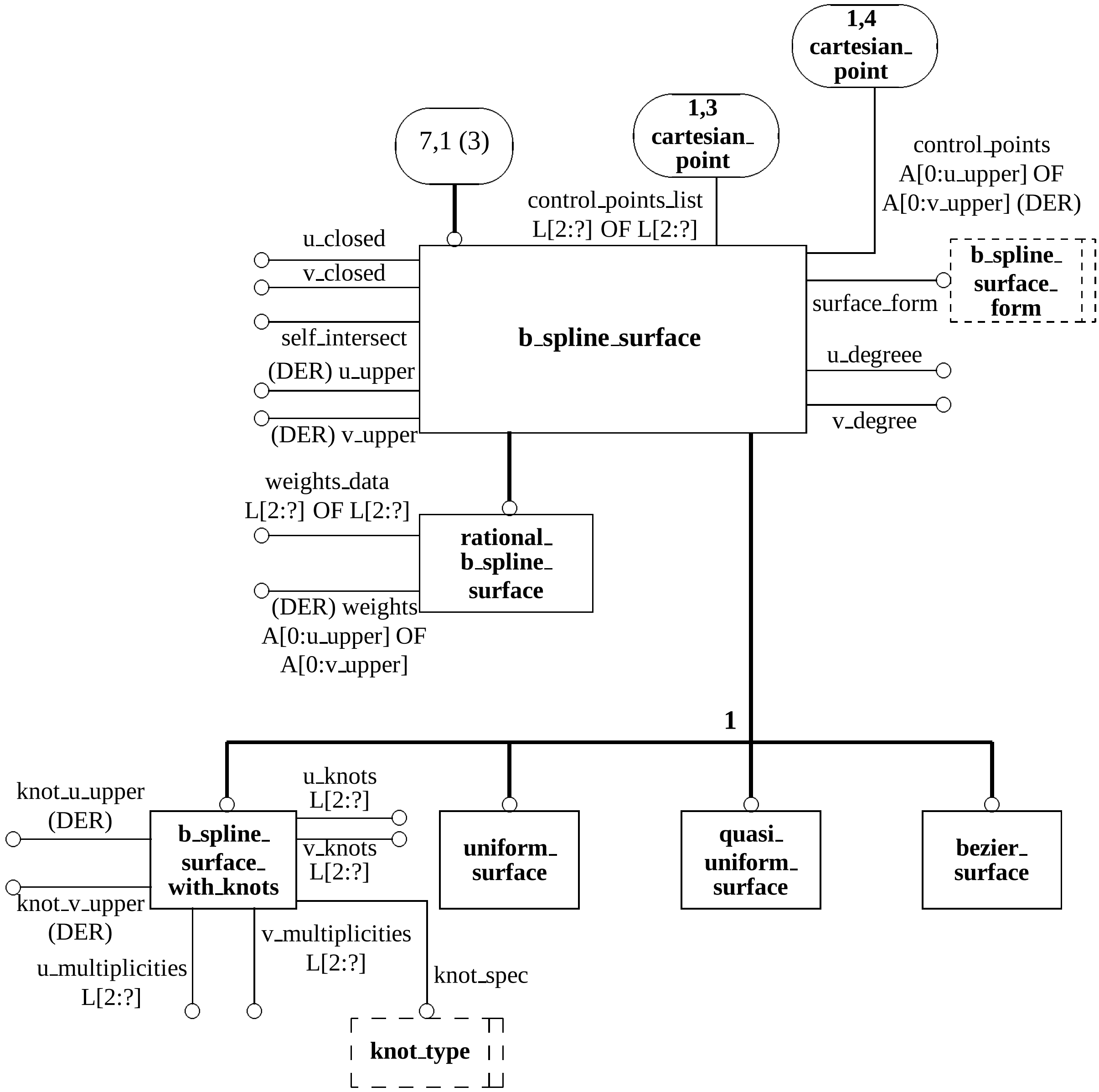
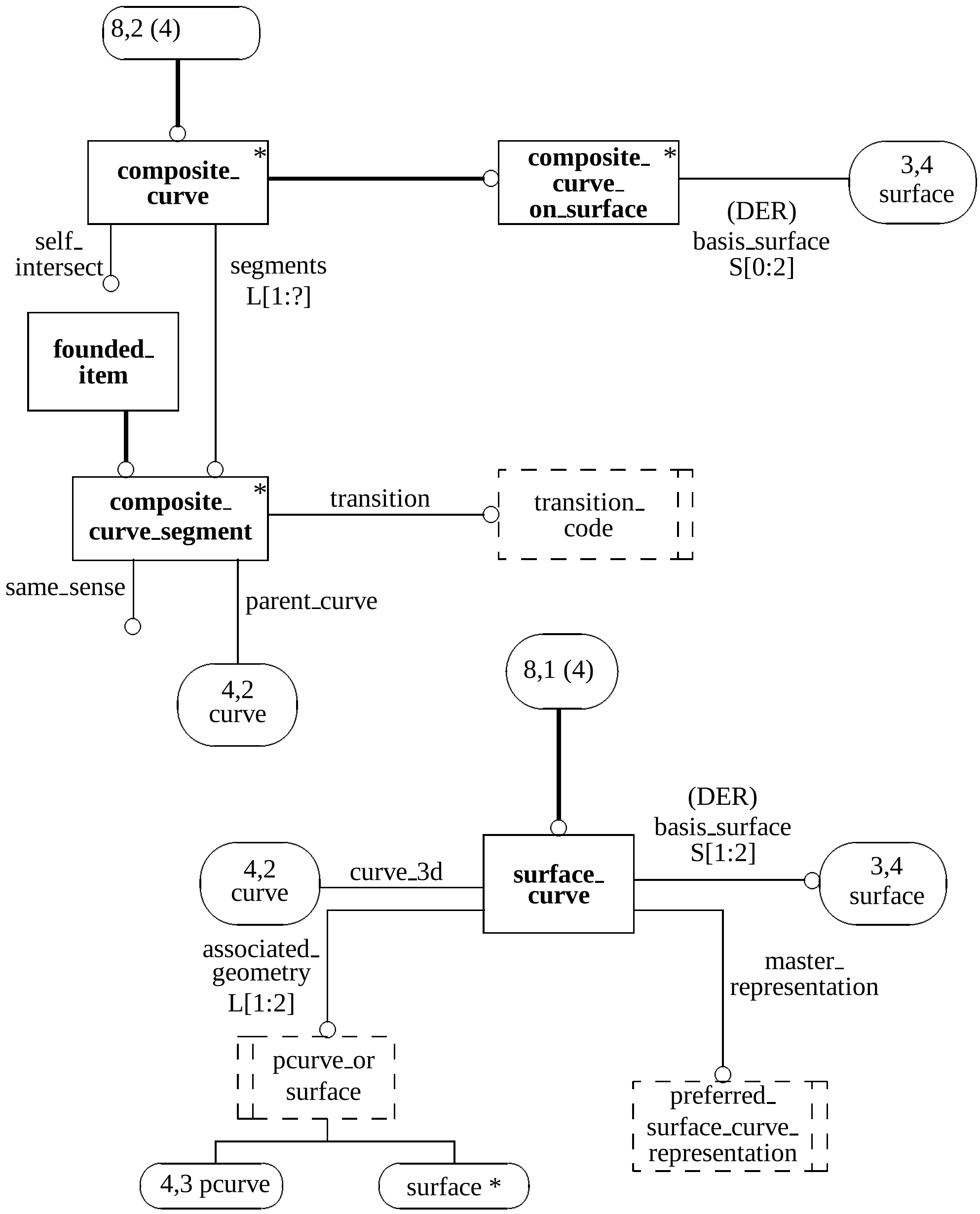


Figure D.7 – aic_manifold_subsurface EXPRESS-G diagram page 7 of 9



* note: excluded by rule on advanced_face

Figure D.8 – aic_manifold_subsurface EXPRESS-G diagram page 8 of 9

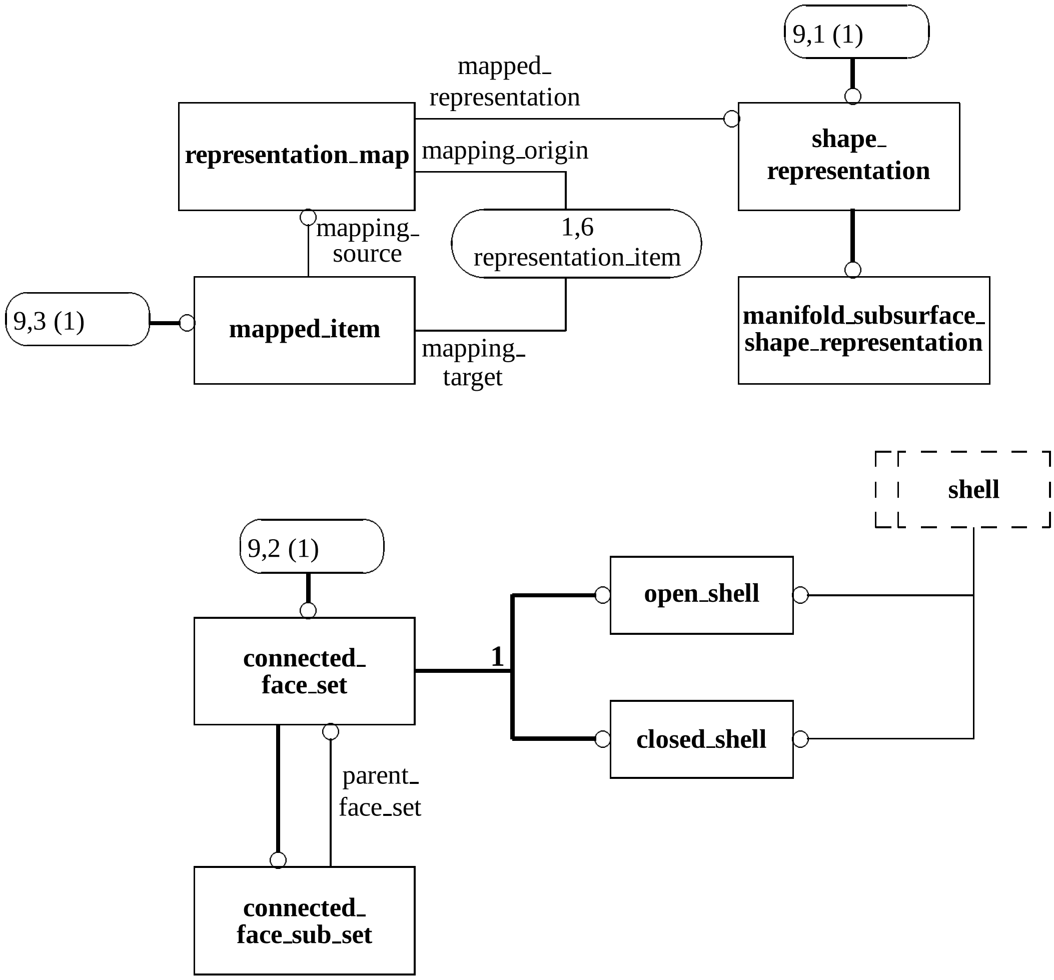


Figure D.9 – aic_manifold_subsurface EXPRESS-G diagram page 9 of 9

Annex E
(informative)

AIC usage example

The part 21 file example below illustrates how this AIC may be used to show the logical relationships between the domains of topological constructs. This is a section of a part 21 file showing all the relevant geometry and topology definitions.

EXAMPLE 1

```
/* Geometry Definition of original closed shell -
 shell is in shape of a cube with half cylinder on top.  */
#1040 =(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#1041 =(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#1100 = CARTESIAN_POINT('origin',(0.0,0.0,0.0));
#1101 = DIRECTION('Dir1',(1.0,0.0,0.0));
#1102 = DIRECTION('Dir2',(0.0,1.0,0.0));
#1103 = DIRECTION('Dir3',(0.0,0.0,1.0));
#1104 = DIRECTION('NegX',(-1.0,0.0,0.0));
/* Points and vertices for face boundaries of closed shell. */
#1105 = VERTEX_POINT('VertPtO',#1100);
#1106 = CARTESIAN_POINT('PtA',(100.0, 0.0, 0.0));
#1107 = VERTEX_POINT('VertPtA',#1106);
#1108 = CARTESIAN_POINT('PtB',(100.0, 100.0, 0.0));
#1109 = VERTEX_POINT('VertPtB',#1108);
#1110 = CARTESIAN_POINT('PtC',(0.0, 100.0, 0.0));
#1111 = VERTEX_POINT('VertPtC',#1110);
#1112 = CARTESIAN_POINT('PtD',(0.0, 0.0 ,100.0));
#1113 = VERTEX_POINT('VertPtD',#1112);
#1114 = CARTESIAN_POINT('PtE',(100.0, 0.0 ,100.0));
#1115 = VERTEX_POINT('VertPtE',#1114);
#1116 = CARTESIAN_POINT('PtF',(100.0, 100.0, 100.0));
#1117 = VERTEX_POINT('VertPtF',#1116);
#1118 = CARTESIAN_POINT('PtG',(0.0, 100.0, 100.0));
#1119 = VERTEX_POINT('VertPtG',#1118);
/* Surfaces for faces */
#1120 = AXIS2_PLACEMENT_3D('Ax2P3DBase',#1100,#1103,#1101);
#1121 = PLANE('Baseplane', #1120);
#1122 = AXIS2_PLACEMENT_3D('Ax2P3DFront',#1100,#1101,#1102);
#1123 = PLANE('Frontplane', #1122);
#1124 = AXIS2_PLACEMENT_3D('Ax2P3DRight',#1100,#1102,#1103);
#1125 = PLANE('Rightplane', #1124);
#1126 = AXIS2_PLACEMENT_3D('Ax2P3DLeft',#1110,#1102,#1103);
#1127 = PLANE('Leftplane', #1126);
#1128 = AXIS2_PLACEMENT_3D('Ax2P3DBack',#1106,#1101,#1102);
#1129 = PLANE('Backplane', #1128);
#1130 = CARTESIAN_POINT('CentreCyl',(50.0, 0.0, 100.0));
#1131 = AXIS2_PLACEMENT_3D('Ax2P3DCyl',#1130,#1102,#1104);
#1132 = CYLINDRICAL_SURFACE('TopCyl',#1131, 50.0);
/* Curves and edge_curves */
#1140 = AXIS2_PLACEMENT_3D('Ax2P3DLcirc',#1154,#1102,#1104);
```

```
#1141 = VECTOR('VecX',#1101, 100.0);
#1142 = VECTOR('VecY',#1102, 100.0);
#1143 = VECTOR('VecZ',#1103, 100.0);
#1144 = LINE('LineOA',#1100, #1141);
#1145 = LINE('LineOC',#1100, #1142);
#1146 = LINE('LineOD',#1100, #1143);
#1147 = LINE('LineAE',#1106, #1143);
#1148 = LINE('LineAB',#1106, #1142);
#1149 = LINE('LineCG',#1110, #1143);
#1150 = LINE('LineCB',#1110, #1141);
#1151 = LINE('LineDG',#1112, #1142);
#1152 = LINE('LineEF',#1114, #1142);
#1153 = CIRCLE('RtCirc',#1131, 50.0);
#1154 = CARTESIAN_POINT('CentreLcirc',(50.0, 100.0, 100.0));
#1155 = LINE('LineBF',#1108,#1143);
#1156 = CIRCLE('LCirc',#1140, 50.0);
#1157 = EDGE_CURVE('EdgeOA',#1105,#1107,#1144,.T.);
#1158 = EDGE_CURVE('EdgeOC',#1105,#1111,#1145,.T.);
#1159 = EDGE_CURVE('EdgeOD',#1105,#1113,#1146,.T.);
#1160 = EDGE_CURVE('EdgeAE',#1107,#1115,#1147,.T.);
#1161 = EDGE_CURVE('EdgeAB',#1107,#1109,#1148,.T.);
#1162 = EDGE_CURVE('EdgeCG',#1111,#1119,#1149,.T.);
#1163 = EDGE_CURVE('EdgeCB',#1111,#1109,#1150,.T.);
#1164 = EDGE_CURVE('EdgeDG',#1113,#1119,#1151,.T.);
#1165 = EDGE_CURVE('EdgeEF',#1115,#1117,#1152,.T.);
#1166 = EDGE_CURVE('EdgeDE',#1113,#1115,#1153,.T.);
#1167 = EDGE_CURVE('EdgeGF',#1119,#1117,#1156,.T.);
#1168 = EDGE_CURVE('EdgeBF',#1109,#1117,#1155,.T.);
/* oriented_edges */
#1169 = ORIENTED_EDGE('OAT',*,*,#1157,.T.);
#1170 = ORIENTED_EDGE('OAF',*,*,#1157,.F.);
#1171 = ORIENTED_EDGE('OCT',*,*,#1158,.T.);
#1172 = ORIENTED_EDGE('OCF',*,*,#1158,.F.);
#1173 = ORIENTED_EDGE('ODT',*,*,#1159,.T.);
#1174 = ORIENTED_EDGE('ODF',*,*,#1159,.F.);
#1175 = ORIENTED_EDGE('AET',*,*,#1160,.T.);
#1176 = ORIENTED_EDGE('AEF',*,*,#1160,.F.);
#1177 = ORIENTED_EDGE('ABT',*,*,#1161,.T.);
#1178 = ORIENTED_EDGE('ABF',*,*,#1161,.F.);
#1179 = ORIENTED_EDGE('CGT',*,*,#1162,.T.);
#1180 = ORIENTED_EDGE('CGF',*,*,#1162,.F.);
#1181 = ORIENTED_EDGE('CBT',*,*,#1163,.T.);
#1182 = ORIENTED_EDGE('CBF',*,*,#1163,.F.);
#1183 = ORIENTED_EDGE('DGT',*,*,#1164,.T.);
#1184 = ORIENTED_EDGE('DGF',*,*,#1164,.F.);
#1185 = ORIENTED_EDGE('EFT',*,*,#1165,.T.);
#1186 = ORIENTED_EDGE('EFF',*,*,#1165,.F.);
#1187 = ORIENTED_EDGE('DET',*,*,#1166,.T.);
#1188 = ORIENTED_EDGE('DEF',*,*,#1166,.F.);
#1189 = ORIENTED_EDGE('GFT',*,*,#1167,.T.);
#1190 = ORIENTED_EDGE('GFF',*,*,#1167,.F.);
#1191 = ORIENTED_EDGE('BFT',*,*,#1168,.T.);
```

```
#1192 = ORIENTED_EDGE('BFF',*,*,#1168,.F.);
/* edge_loops */
#1201 = EDGE_LOOP('ELOCBA',(#1171, #1181, #1178, #1170));
#1202 = EDGE_LOOP('ELOAED',(#1169, #1175, #1188, #1174));
#1203 = EDGE_LOOP('ELODGC',(#1173, #1183, #1180, #1172));
#1204 = EDGE_LOOP('ELABFE',(#1177, #1191, #1186, #1176));
#1205 = EDGE_LOOP('ELCGFB',(#1179, #1189, #1192, #1182));
#1206 = EDGE_LOOP('ELDEFG',(#1187, #1185, #1190, #1184));
/* face_bounds and advanced_faces */
#1211 = FACE_OUTER_BOUND('baseBd',#1201,.T.);
#1212 = FACE_OUTER_BOUND('rightBd',#1202,.T.);
#1213 = FACE_OUTER_BOUND('frontBd',#1203,.T.);
#1214 = FACE_OUTER_BOUND('backBd',#1204,.T.);
#1215 = FACE_OUTER_BOUND('leftBd',#1205,.T.);
#1216 = FACE_OUTER_BOUND('TopcylBd',#1206,.T.);
#1221 = ADVANCED_FACE('BaseFace',(#1211),#1121,.F.);
#1222 = ADVANCED_FACE('RightFace',(#1212),#1125,.F.);
#1223 = ADVANCED_FACE('FrontFace',(#1213),#1123,.F.);
#1224 = ADVANCED_FACE('BackFace',(#1214),#1129,.T.);
#1225 = ADVANCED_FACE('LeftFace',(#1215),#1127,.T.);
#1226 = ADVANCED_FACE('TopcylFaceO',(#1216),#1132,.T.);
/* closed_shell */
#1250 = CLOSED_SHELL('CubeCyl', (#1221, #1222, #1223, #1224, #1225, #1226));
/* New point and vertex for subset1, M is 1/3 of way along semi-circle GF */
#1300 = CARTESIAN_POINT('PtM',(25.0, 100.0, 143.3012702));
#1301 = VERTEX_POINT('VertPtM', #1300);
/* Edge DM is defined as a surface curve on cylindrical face via a pcurve.
 Define 2D context and 2D geometry for pcurve (line in parameter space) */
#1302 = (GEOMETRIC_REPRESENTATION_CONTEXT(2)
             PARAMETRIC_REPRESENTATION_CONTEXT()
             REPRESENTATION_CONTEXT('CylSurf','Parameter_space'));
#1303 = CARTESIAN_POINT('PtOparam',(0.0, 0.0));
#1304 = DIRECTION('Dir2D', (1.047197551, 100.0));
#1305 = VECTOR('Vec2D', #1304, 100.013708);
#1306 = LINE('LinPcrv', #1303, #1305);
#1307 = DEFINITIONAL_REPRESENTATION('Pcurvrep', (#1306), #1302);
#1308 = PCURVE('CylPcrv', #1132, #1307);
/* Define approximate 3D geometry of surface curve D to M */
#1310 = CARTESIAN_POINT('P2', (0.0, 33.33333333, 117.4532952));
#1311 = CARTESIAN_POINT('P3', (9.885005297, 66.666666667, 134.5746238));
#1312 = BEZIER_CURVE('CylCrv3D', 3, (#1112, #1310, #1311, #1300),
           .UNSPECIFIED., .F., .F.);
#1313 = SURFACE_CURVE('CrvBM3D', #1312, (#1308), .PCURVE_S1.);
/* Define new edges for subset 1. */
#1321 = EDGE_CURVE('EdgeDM', #1113, #1301, #1313, .T. );
#1322 = ORIENTED_EDGE('DMT', *, *, #1321, .T.);
#1323 = SUBEDGE('EdgeGM', #1119, #1301, #1167);
#1324 = ORIENTED_EDGE('GMF', *, *, #1323, .F.);
/* Define subface and subset 1 (as cfss and open_shell). */
#1325 = EDGE_LOOP('ELDMG', (#1322, #1324, #1184));
#1326 = FACE_OUTER_BOUND('SubCylFac1Bd', #1325, .T.);
#1327 = SUBFACE('SubCylF1', (#1326), #1236);
```

```
#1350 = (CONNECTED_FACE_SET( (#1327, #1223))
        CONNECTED_FACE_SUB_SET(#1250)
        OPEN_SHELL( )
        REPRESENTATION_ITEM('Subset1')
        TOPOLOGICAL_REPRESENTATION_ITEM( ));
/* Define new edges and associated geometry for subset 2 (inside Subset1). */
#1400 = CARTESIAN_POINT('PtP',(0.0, 65.0, 50.0));
#1401 = VERTEX_POINT('VertPtP', #1400);
#1402 = CARTESIAN_POINT('PtQ',(0.0, 65.0, 100.0));
#1403 = VERTEX_POINT('VertPtQ', #1402);
#1404 = CARTESIAN_POINT('PtR',(10.0, 65.0, 130.0));
#1405 = VERTEX_POINT('VertPtR', #1404);
#1406 = CARTESIAN_POINT('PtS',(10.0, 90.0, 130.0));
#1407 = VERTEX_POINT('VertPtS', #1406);
#1408 = CARTESIAN_POINT('PtT',(0.0, 90.0, 100.0));
#1409 = VERTEX_POINT('VertPtT', #1408);
#1410 = CARTESIAN_POINT('PtU',(0.0, 90.0, 50.0));
#1411 = VERTEX_POINT('VertPtU', #1410);
#1412 = LINE('LinePQ', #1400, #1143);
#1413 = LINE('LinePU', #1400, #1142);
#1414 = LINE('LineRS', #1404, #1142);
#1415 = LINE('LineUT', #1410, #1143);
#1416 = CARTESIAN_POINT('CentreCirc2',(50.0, 65.0, 100.0));
#1417 = AXIS2_PLACEMENT_3D('Ax2P3DCirc2',#1416,#1102,#1104);
#1418 = CIRCLE('Circ2',#1417, 50.0);
#1419 = CARTESIAN_POINT('CentreCirc3',(50.0, 90.0, 100.0));
#1420 = AXIS2_PLACEMENT_3D('Ax2P3DCirc3',#1419,#1102,#1104);
#1421 = CIRCLE('Circ3',#1420, 50.0);
#1422 = EDGE_CURVE('EdgePQ', #1401, #1403, #1412, .T.);
#1423 = EDGE_CURVE('EdgePU', #1401, #1411, #1413, .T.);
#1424 = EDGE_CURVE('EdgeRS', #1405, #1407, #1414, .T.);
#1425 = EDGE_CURVE('EdgeUT', #1411, #1409, #1415, .T.);
#1426 = EDGE_CURVE('EdgeQR', #1403, #1405, #1418, .T.);
#1427 = EDGE_CURVE('EdgeTS', #1409, #1407, #1421, .T.);
#1428 = SUBEDGE('EdgeQT', #1403, #1409, #1164);
/* Define edge_loops, face_bounds and subfaces */
#1429 = ORIENTED_EDGE('PQT', *, *, #1422, .T.);
#1430 = ORIENTED_EDGE('PUF', *, *, #1423, .F.);
#1431 = ORIENTED_EDGE('RST', *, *, #1424, .T.);
#1432 = ORIENTED_EDGE('UTF', *, *, #1425, .F.);
#1433 = ORIENTED_EDGE('QRT', *, *, #1426, .T.);
#1434 = ORIENTED_EDGE('TSF', *, *, #1427, .F.);
#1435 = ORIENTED_EDGE('QTT', *, *, #1428, .T.);
#1436 = ORIENTED_EDGE('QTF', *, *, #1428, .F.);
#1437 = EDGE_LOOP('ELPQTU', (#1429, #1435, #1432, #1430));
#1438 = EDGE_LOOP('ELQRST', (#1433, #1431, #1434, #1436));
#1439 = FACE_OUTER_BOUND('SubCylFac2Bd', #1438, .T.);
#1440 = FACE_OUTER_BOUND('SubFrontBd', #1437, .T.);
#1441 = SUBFACE('SubCylF2', (#1439), #1327);
#1442 = SUBFACE('SubFront', (#1440), #1223);
#1450 = (CONNECTED_FACE_SET( (#1441, #1442))
        CONNECTED_FACE_SUB_SET(#1350)
```

```
        OPEN_SHELL( )
        REPRESENTATION_ITEM('Subset2')
        TOPOLOGICAL_REPRESENTATION_ITEM( ));
#1490 = (GEOMETRIC_REPRESENTATION_CONTEXT(3)
        GLOBAL_UNIT_ASSIGNED_CONTEXT((#1040,#1041))
        REPRESENTATION_CONTEXT('Context for Subsets',
        'This is a 3D context using millimetres'));
#1500 = MANIFOLD_SUBSURFACE_SHAPE_REPRESENTATION('SubsetRep',
            (#1350, #1450), #1490);
```

NOTE 1 #1250 is a **closed_shell** in the form of a cube with a half cylinder on top. It has 6 faces, 5 of them planar, with a cylindrical top face. This could be used to define an **advanced_brep_shape_representation** from part 514, or a **manifold_surface_shape_representation** from part 509. For the purpose of naming the faces and describing the geometry it is assumed that the closed shell is being viewed from a point along the negative X axis with the Z axis pointing towards the top of the cylindrical face.

NOTE 2 #1350 defines an instance of **open_shell** and **connected_face_sub_set** that refers to #1250 as its **parent_face_set**. It consists of 2 faces, one of these is the front face from #1250 and the second is a **subface** of the cylindrical top face of #1250. One of the edges of this triangular **subface** is a **subedge**, one is an existing edge and the third has its geometry defined by a **pcurve** on the cylindrical surface. This **pcurve** is a line in parameter space from the point (0,0) to ($\pi/3$, 100). A Bezier curve provides a slightly less precise 3 dimensional representation of this **surface_curve**.

NOTE 3 #1450 illustrates the possibility of nesting **connected_face_sub_set**s. It is defined with #1350 as **parent_face_set**. It consists of two **subface**s, each related to a face of #1350. The shared edge between these **subfaces** is another example of a **subedge**.

NOTE 4 #1500 is an instance of **manifold_subsurface_shape_representation** containing **connected_face_sub_set**s #1350 and #1450. It is defined in a 3 dimensional **geometric_representation_context** using millimetres and radians as units. This provides the context for all 3D geometry in the file. The geometry defined in this file is illustrated in figure E.1.

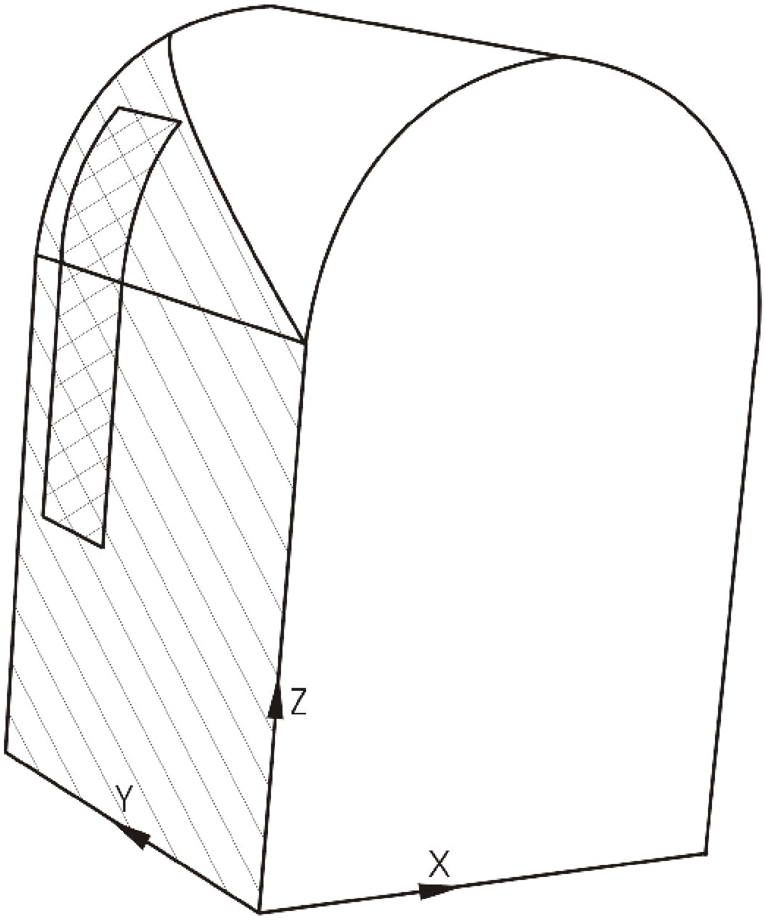


Figure E.1 – Manifold_subsurface_shape_representation – faces and subfaces

Bibliography

[1] ISO 10303-21 *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*

[2] ISO 10303-509 *Industrial automation systems and integration — Product data representation and exchange — Part 509: Application interpreted construct: Manifold surface*

[3] ISO 10303-514 *Industrial automation systems and integration — Product data representation and exchange — Part 514: Application interpreted construct: Advanced boundary representation*

Index

ICS 25.040.40

Price based on 36 pages