INTERNATIONAL STANDARD

ISO 10303-510

First edition
2000-03-01

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

Part 510: Application interpreted construct: Geometrically bounded wireframe

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

Partie 510: Construction interprétée d'application: Cadre en fil métallique délimité géométriquement

Reference number
ISO 10303-510:2000(E)

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.

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.

International Standard ISO 10303-501 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*.

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration - Product data representation and exchange*:

— Part 1, Overview and fundamental principles;

— Part 11, Description methods: The EXPRESS language reference manual;

— Part 12, Description method: The EXPRESS-I language reference manual;

— Part 21, Implementation methods: Clear text encoding of the exchange structure;

— Part 22, Implementation method: Standard data access interface specification;

— Part 23, Implementation method: C++ language binding to the standard data access interface;

— Part 24, Implementation method: C language binding to the standard data access interface;

— Part 26, Implementation method: Interface definition language binding to the standard data access;

— Part 31, Conformance testing methodology and framework: General concepts;

— Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;

— Part 34, Conformance testing methodology and framework: Abstract test methods;

— Part 35, Conformance testing methodology and framework: Abstract test methods for SDAI implementations;

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

— Part 42, Integrated generic resources: Geometric and topological representation;

— Part 43, Integrated generic resources: Representation structures;

— Part 44, Integrated generic resources: Product structure configuration;

— Part 45, Integrated generic resource: Materials;

— Part 46, Integrated generic resources: Visual presentation;

— Part 47, Integrated generic resource: Shape variation tolerances;

— Part 49, Integrated generic resource: Process structure and properties;

— Part 101, Integrated application resource: Draughting;

— Part 104, Integrated application resource: Finite element analysis;

— Part 105, Integrated application resource: Kinematics;

— Part 106, Integrated application resource: Building construction core model;

— Part 107, Engineering Analysis Core Application reference model (EA C-ARM);

— Part 108, Integrated application resource: Parameterization and constraints for explicit geometric product models

— Part 201, Application protocol: Explicit draughting;

— Part 202, Application protocol: Associative draughting;

— Part 203, Application protocol: Configuration controlled design;

— Part 204, Application protocol: Mechanical design using boundary representation;

— Part 205, Application protocol: Mechanical design using surface representation;

— Part 207, Application protocol: Sheet metal die planning and design;

— Part 208, Application protocol: Life cycle management - Change process;

— Part 209, Application protocol: Composite and metallic structural analysis and related design;

— Part 210, Application protocol: Electronic assembly, interconnet, and packaging design;

— Part 212, Application protocol: Electrotechnical design and installation

— Part 213, Application protocol: Numerical control process plans for machined parts;

— Part 214, Application protocol: Core data for automotive mechanical design processes;

— Part 215, Application protocol: Ship arrangement;

— Part 216, Application protocol: Ship moulded forms;

— Part 217, Application protocol: Ship piping;

— Part 218, Application protocol: Ship structures;

— Part 220, Application protocol: Process planning, manufacture, and assembly of layered electronic products

— Part 221, Application protocol: Functional data and their schematic representation for process plant;

— Part 222, Application protocol: Exchange of product data for composite structures;

— Part 223, Application protocol: Exchange of design and manufacturing product information for casting parts;

— Part 224, Application protocol: Mechanical product definition for process plans using machining features;

— Part 225, Application protocol: Building elements using explicit shape representation;

— Part 226, Application protocol: Ship mechanical systems;

— Part 227, Application protocol: Plant spatial configuration;

— Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;

— Part 230, Application protocol: Building structural frame: Steelwork;

— Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;

— Part 232, Application protocol: Technical data packaging core information and exchange;

— Part 233, Application Protocol: Systems engineering data representation

— Part 234, Application protocol: Ship Operational logs, records, and messages

— Part 235, Application Protocol: Materials information for the design and verification of products

— Part 301, Abstract test suite: Explicit draughting;

— Part 302, Abstract test suite: Associative draughting;

— Part 303, Abstract test suite: Configuration controlled design;

— Part 304, Abstract test suite: Mechanical design using boundary representation;

— Part 305, Abstract test suite: Mechanical design using surface representation;

— Part 307, Abstract test suite: Sheet metal die planning and design;

— Part 308, Abstract test suite: Life cycle management - Change process;

— Part 309, Abstract test suite: Composite and metallic structural analysis and related design;

— Part 310, Abstract test suite: Electronic assembly, interconnect, and packaging design;

— Part 312, Abstract test suite: Electrotechnical design and installation;

— Part 313, Abstract test suite: Numerical control process plans for machined parts;

— Part 314, Abstract test suite: Core data for automotive mechanical design processes;

— Part 315, Abstract test suite: Ship arrangement;

— Part 316, Abstract test suite: Ship moulded for

— Part 317, Abstract test suite: Ship piping;

— Part 318, Abstract test suite: Ship structures;

— Part 321, Abstract test suite: Functional data and their schematic representation for process plant;

— Part 322, Abstract test suite: Exchange of product data for composite structures;

— Part 323, Abstract test suite: Exchange of design and manufacturing product information for casting parts;

— Part 324, Abstract test suite: Mechanical product definition for process plans using machining features;

— Part 325, Abstract test suite: Building elements using explicit shape representation;

— Part 326, Abstract test suite: Ship mechanical systems;

— Part 327, Abstract test suite: Plant spatial configuration;

— Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;

— Part 330, Abstract test suite: Building structural frame: Steelwork;

— Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;

— Part 332, Abstract test suite: Technical data packaging core information and exchange;

— Part 334, Abstract test suitel: Ship Operational logs, records, and messages

— Part 335, Abstract test suite: Materials information for the design and verification of products

— Part 501, Application interpreted construct: Edge-based wireframe;

— Part 502, Application interpreted construct: Shell-based wireframe;

— Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;

— Part 504, Application interpreted construct: Draughting annotation;

— Part 505, Application interpreted construct: Drawing structure and administration;

— Part 506, Application interpreted construct: Draughting elements;

— Part 507, Application interpreted construct: Geometrically bounded surface;

— Part 508, Application interpreted construct: Non-manifold surface;

— Part 509, Application interpreted construct: Manifold surface;

— Part 510, Application interpreted construct: Geometrically bounded wireframe;

— Part 511, Application interpreted construct: Topologically bounded surface;

— Part 512, Application interpreted construct: Faceted boundary representation;

— Part 513, Application interpreted construct: Elementary boundary representation;

— Part 514, Application interpreted construct: Advanced boundary representation;

— Part 515, Application interpreted construct: Constructive solid geometry;

— Part 517, Application interpreted construct: Mechanical design geometric presentation;

— Part 518, Application interpreted construct: Mechanical design shaded presentation;

— Part 519, Application interpreted construct: Geometric tolerances;

— Part 520, Application interpreted construct: Associative draughting.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of the International Standard reflects its structure:

— Parts 11 to 12 specify the description methods,

— Parts 21 to 26 specify the implementation methods,

— Parts 31 to 35 specify the conformance testing methodology and framework,

— Parts 41 to 49 specify the integrated generic resources,

— Parts 101 to 108 specify the integrated application resources,

— Parts 201 to 235 specify the application protocols,

— Parts 301 to 335 specify the abstract test suites, and

— Parts 501 to 520 specify the application interpreted constructs.

Should further parts be published, they will follow the same numbering pattern.

Annexes A and B form an integral part of this part of ISO 10303. Annex C and D are for information only.

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 product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the application interpreted constructs 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 the description of a geometric shape by means of geometrically bounded wireframe. It includes the geometric entities for the definition of elementary and complex bounded curves. It does not include the definition of geometric surfaces or topological connectivity of the geometry.

Industrial automation systems and integration — Product data representation and exchange — Part 510: Application interpreted construct: Geometrically bounded wireframe

1 Scope

This part of ISO 10303 specifies the interpretation of the integrated resources to satisfy requirements for the representation of the product shape using geometrically bounded wireframe models.

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

— points and curve geometry in 3D cartesian space;

— the use of wireframe geometry to represent a shape;

— the combination of representations to form an aggregation of representations.

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

— surface geometry;

— topological connectivity of geometry;

— product definitions.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 8824-1:1995, *Information technology - Open systems interconnection - 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 10303-41:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resources: Fundamentals of product description and support*.

ISO 10303-42:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resources: Geometric and topological representation*.

ISO 10303-43:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resources: Representation structures*.

ISO 10303-202:1996, *Industrial automation systems and integration — Product data representation and exchange — Part 202: Application protocol: Associative draughting.*

3 Definitions and abbreviations

3.1 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms defined in ISO 10303-1:

— application;

— application context;

— application protocol;

— implementation method;

— integrated resource;

— interpretation;

— model;

— product;

— product data.

3.2 Terms defined in ISO 10303-202

This part of ISO 10303 makes use of the following terms defined in ISO 10303-202:

— Application interpreted construct (AIC).

3.3 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.

EXPRESS Specification:

```
*)
SCHEMA aic_geometrically_bounded_wireframe;

  USE FROM geometric_model_schema                         -- ISO 10303-42
    (geometric_curve_set,
     geometric_set);

  USE FROM geometry_schema                                -- ISO 10303-42
    (axis1_placement,
     axis2_placement_3d,
     b_spline_curve_with_knots,
     bezier_curve,
     cartesian_transformation_operator_3d,
     circle,
     composite_curve,
```

```
      conic,
      curve,
      curve_replica,
      ellipse,
      geometric_representation_context,
      hyperbola,
      line,
      offset_curve_3d,
      parabola,
      point,
      point_on_curve,
      point_replica,
      polyline,
      quasi_uniform_curve,
      rational_b_spline_curve,
      reparametrised_composite_curve_segment,
      trimmed_curve,
      uniform_curve);

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

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

NOTES

2 - The schemas referenced above can be found in the following Parts of ISO 10303:

geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
product_property_representation_schema	ISO 10303-41
representation_schema	ISO 10303-43

3 - The entity **conic** is explicitly interfaced (i.e. included in the USE FROM lists) to allow rules within the **geometrically_bounded_wireframe_shape_representation** entity to access attributes of this entity. For the use of this AIC, this entity shall only be instantiated as one of its subtypes.

4.1 Introduction

This part of ISO 10303 provides a set of geometric and wireframe model entities for the representation of a shape that consists of points and curves. The curves are bounded and represented by the entity **geometrically_bounded_wireframe_shape_representation**, that is a **shape_representation** (see ISO 10303-41).

4.2 Fundamental concepts and assumptions

The wireframe representation of a shape is a collection of points and curves that are associated with one another to form a **wireframe_model**. All geometry is part of the shape representation. There is no "loose" or "unconnected" geometry in the wireframe model.

The shapes represented by the **geometrically_bounded_wireframe_shape_representation** are assumed to be representable by wireframe geometry. In other words, this AIC may not be used to represent shapes with complex surfaces (such as airfoils).

Wireframe representations may be assemblies of other wireframe representations that are mapped to the same coordinate space.

4.3 aic_geometrically_bounded_wireframe entity definition: geometrically_-bounded_wireframe_shape_representation

A **geometrically_bounded_wireframe_shape_representation** is a **shape_representation** that represents the shape or portions of the shape of a **product** by wireframe geometry without topology. These representations are formed by the use of points and curves only. All unbounded curves shall be explicitly trimmed unless they are closed. The boundaries of the curves shall be defined explicitly by points on the curves and explicit associations between the points and the curves that they bound or by parameter values. Each **geometric_set** in a **geometrically_bounded_wireframe_shape_representation** shall contain only those entities that define the physical object that is being represented by a particular instance of the **geometrically_bounded_wireframe_shape_representation**. The geometric entities that are used to support the definition of another geometric entity shall not exist themselves in the **elements** set of a **geometric_set**.

NOTE - An application protocol that uses this AIC may ensure that the **shape_representation** entity is instantiated as an **geometrically_bounded_wireframe_shape_representation**.

EXAMPLE - A circular arc is to be used to define the corner radius on a part that is being represented using a **geometrically_-bounded_wireframe_shape_representation**. The representation of the arc is a **trimmed_curve** whose **basis_curve** is a **circle.**

<u>EXPRESS specification</u>:

```
*)
ENTITY geometrically_bounded_wireframe_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (it <* SELF.items |
         NOT (SIZEOF (TYPEOF(it) *
           ['AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.GEOMETRIC_CURVE_SET',
            'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.AXIS2_PLACEMENT_3D',
            'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.MAPPED_ITEM']) = 1)
            )) = 0;
  WR2: SIZEOF (QUERY (it <* SELF.items |
         SIZEOF (TYPEOF (it) *
           ['AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.GEOMETRIC_CURVE_SET',
            'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.MAPPED_ITEM']) = 1
            )) >= 1;
  WR3: SIZEOF (QUERY (gcs <* QUERY (it <* SELF.items |
         ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.GEOMETRIC_CURVE_SET'
         IN TYPEOF (it))) | NOT (SIZEOF (QUERY (crv <*
         QUERY (elem <* gcs\geometric_set.elements |
         ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.CURVE' IN TYPEOF
        (elem))) |
```

```
          NOT (valid_geometrically_bounded_wf_curve(crv))
          )) = 0)
          )) = 0;
  WR4: SIZEOF (QUERY (gcs <* QUERY (it <* SELF.items |
          ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.GEOMETRIC_CURVE_SET'
          IN TYPEOF (it))) |NOT (SIZEOF (QUERY (pnts <*
          QUERY (elem <* gcs\geometric_set.elements |
            ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.POINT' IN TYPEOF (elem)))|
            NOT (valid_geometrically_bounded_wf_point(pnts))
            )) = 0)
            )) = 0;
  WR5: SIZEOF (QUERY (gcs <* QUERY (it <* SELF.items |
          ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.GEOMETRIC_CURVE_SET'
          IN TYPEOF (it))) |NOT (SIZEOF (QUERY (cnc <*
          QUERY (elem <* gcs\geometric_set.elements |
          ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.CONIC' IN TYPEOF (elem)))
          | NOT('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.AXIS2_PLACEMENT_3D'
          IN TYPEOF (cnc\conic.position))
          )) = 0)
          )) = 0;
  WR6: SIZEOF (QUERY (gcs <* QUERY (it <* SELF.items |
          ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.GEOMETRIC_CURVE_SET'
          IN TYPEOF (it))) | NOT (SIZEOF (QUERY (pline <*
          QUERY (elem <* gcs\geometric_set.elements |
          ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.POLYLINE' IN TYPEOF (elem))) |
          NOT (SIZEOF (pline\polyline.points) > 2)
          )) = 0)
          )) = 0;
  WR7: SIZEOF (QUERY (mi <* QUERY (it <* SELF.items |
         ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.MAPPED_ITEM'
         IN TYPEOF (it))) |NOT ('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.' +
         'GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION'
         IN TYPEOF (mi\mapped_item.mapping_source.mapped_representation)))
         )= 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **items** in a **geometrically_bounded_wireframe_shape_representation** shall be a **geometric_-curve_set**, **axis2_placement_3d**, or **mapped_item** .

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

WR3: Every **offset_curve** in a **geometric_curve_set** of a **geometrically_bounded_wireframe_shape_-representation** shall have a basis curve that is a **polyline**, **b_spline_curve**, **ellipse**, or **circle**. Every **curve_replica** in a **geometric_curve_set** of a **geometrically_bounded_wireframe_shape_representation** shall have a parent curve that is a **polyline**, **b_spline_curve**, **ellipse**, or **circle**. Every **composite_curve** in a **geometric_curve_set** of a **geometrically_bounded_wireframe_shape_representation** shall have segments that can be **polyline**, **b_spline_curve**, **ellipse**, or **circle.** Every **trimmed_curve** in a **geometric_curve_set** of a **geometrically_bounded_wireframe_shape_representation** shall have a basis curve that is either a **polyline**, **b_spline_curve**, **ellipse**, **circle**,**line**, **parabola**, or **hyperbola**.

WR4: All **point**s that are in the **elements** of the **geometric_curve_set** for a **geometrically_bounded_-wireframe_shape_representation** shall be a **cartesian_point**, **point_on_curve**, or **point_replica**. The **point_-replica** shall replicate either another **point_replica** or a **cartesian_point**. The **point_on_curve** shall lie on a valid curve type for a **geometrically_bounded_wireframe_shape_representation**.

WR5: The **position** for a **conic** in the **elements** of a **geometric_curve_set** for a **geometrically_bounded_-wireframe_shape_representation** shall only be an **axis2_placement_3d**.

WR6: Every **polyline** in the **elements** of a **geometric_curve_set** for a **geometrically_bounded_wireframe_-shape_representation** shall contain more than two distinct **point**s.

WR7: If there is a **mapped_item** in a **geometrically_bounded_wireframe_shape_representation**, the source of the **mapped_item** shall be a **geometrically_bounded_wireframe_shape_representation**.

4.4 aic_geometrically_bounded_wireframe function definitions

4.4.1 valid_geometrically_bounded_wf_curve

The **valid_geometrically_bounded_wf_curve** function determines whether or not an input curve is valid for use in representing a shape defined by a geometrically bounded wireframe. It is concerned with the correct use of bounded curves to serve as reference curves for offset_curve, curve_replica, and composite_curve. If an unbounded curve such as a parabola or hyperbola is used, it must be trimmed. The function is recursive in its structure in order to check to the necessary levels.

<u>EXPRESS specification</u>:

```
*)
FUNCTION valid_geometrically_bounded_wf_curve
   (crv: curve) : BOOLEAN;

  -- check for valid basic curve types
  IF SIZEOF (['AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.POLYLINE',
             'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.B_SPLINE_CURVE',
             'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.ELLIPSE',
             'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.CIRCLE'] * TYPEOF (crv)) =    1
    THEN RETURN (TRUE);
  ELSE
    -- if the curve is a trimmed_curve
    IF (('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.TRIMMED_CURVE') IN TYPEOF (crv))
THEN
      -- if a line, parabola, or hyperbola is being trimmed, then valid
      IF SIZEOF (['AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.LINE',
                  'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.PARABOLA',
                  'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.HYPERBOLA'] *
```

```
          TYPEOF(crv\trimmed_curve.basis_curve)) = 1
        THEN RETURN (TRUE);

      -- otherwise, recursively check basis_curve
      ELSE RETURN (valid_geometrically_bounded_wf_curve
                   (crv\trimmed_curve.basis_curve));
      END_IF;
    ELSE
      -- recursively check the offset_curve basis curve
      IF (('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.OFFSET_CURVE_3D')
          IN TYPEOF (crv))
        THEN RETURN (valid_geometrically_bounded_wf_curve
                     (crv\offset_curve_3d.basis_curve));
      ELSE
        -- recursively check the curve_replica parent curve
        IF (('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.CURVE_REPLICA')
             IN TYPEOF (crv))
          THEN RETURN (valid_geometrically_bounded_wf_curve
                     (crv\curve_replica.parent_curve));
        ELSE
          -- recursively check the composite_curve segments
          IF (('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.COMPOSITE_CURVE')
              IN  TYPEOF (crv)) THEN
            RETURN (SIZEOF (QUERY (ccs <* crv\composite_curve.segments |
                  NOT (valid_geometrically_bounded_wf_curve
                       (ccs.parent_curve)))) = 0);
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);
END_FUNCTION;
(*
```

Argument definitions:

crv: the input **curve** that is to be examined.

4.4.2 valid_geometrically_bounded_wf_point

The **valid_geometrically_bounded_wf_point** function determines whether or not an input point is valid for use in representing a shape defined by a geometrically bounded wireframe. It is concerned with ensuring that any point_on_curve must lie on a bounded curve type and this is done by using the previous function: valid_geometrically_bounded_wf_curve. The function has a recursive structure in order to check at different levels.

EXPRESS specification:

```
*)
FUNCTION valid_geometrically_bounded_wf_point
   (pnt : point) : BOOLEAN;
```

```
  -- check for valid basis types
  IF (( 'AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.CARTESIAN_POINT') IN TYPEOF (pnt))
    THEN RETURN (TRUE);
  ELSE
    -- if the input type is a point_on_curve then check for a valid
    -- geometrically bounded curve type as the basis
    IF (('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.POINT_ON_CURVE') IN TYPEOF (pnt))
      THEN RETURN (valid_geometrically_bounded_wf_curve
                      (pnt\point_on_curve.basis_curve));

    ELSE
      -- if the input type is a point_replica check for a valid parent point
      IF (('AIC_GEOMETRICALLY_BOUNDED_WIREFRAME.POINT_REPLICA') IN TYPEOF (pnt))
        THEN RETURN (valid_geometrically_bounded_wf_point
                          (pnt\point_replica.parent_pt));
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);
END_FUNCTION;
(*
```

Argument definitions:

pnt: the input **point** that is to be examined.

```
*)
END_SCHEMA; -- aic_geometrically_bounded_wireframe
(*
```

Annex A
(normative)

Short names of entities

Table A.1 provides the short names of entities specified in 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 names	Short names
GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION	GBWSR

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(510) 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-geometrically-bounded-wireframe-schema in an open system, the object identifier

{ iso standard 10303 part(510) version(1) object(1) aic-geometrically-bounded-wireframe-schema(1) }

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

Annex C
(informative)

EXPRESS-G diagrams

Figure C.1 through C.6 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 - The following select types: transformation and vector_or_direction, are interfaced into the AIC expanded listing according to the implicit interface rules of ISO 10303-11. These select types are not referenced by other entities in this part of ISO 10303.

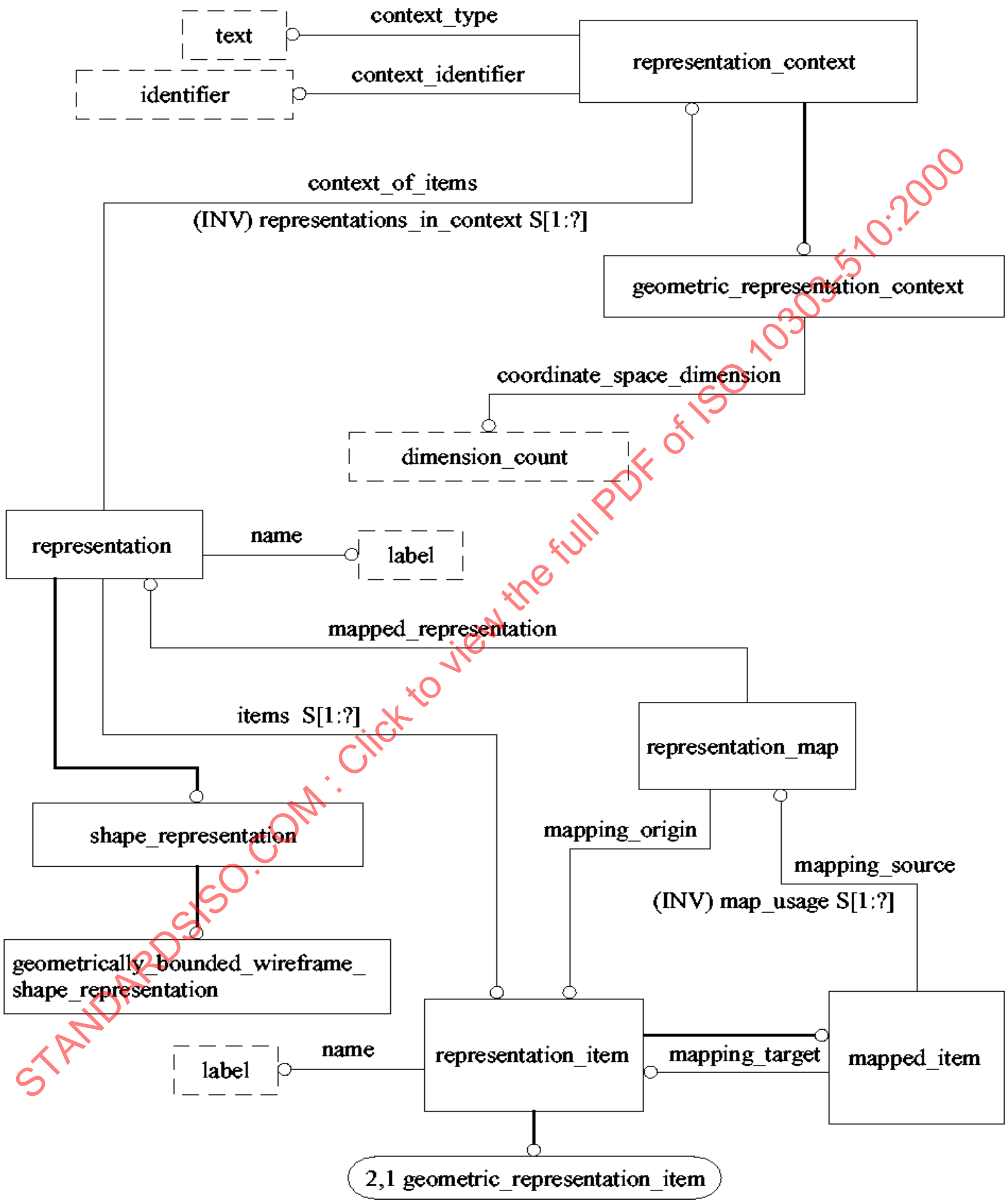

Figure C.1 - aic_geometrically_bounded_wireframe - EXPRESS-G diagram 1 of 6

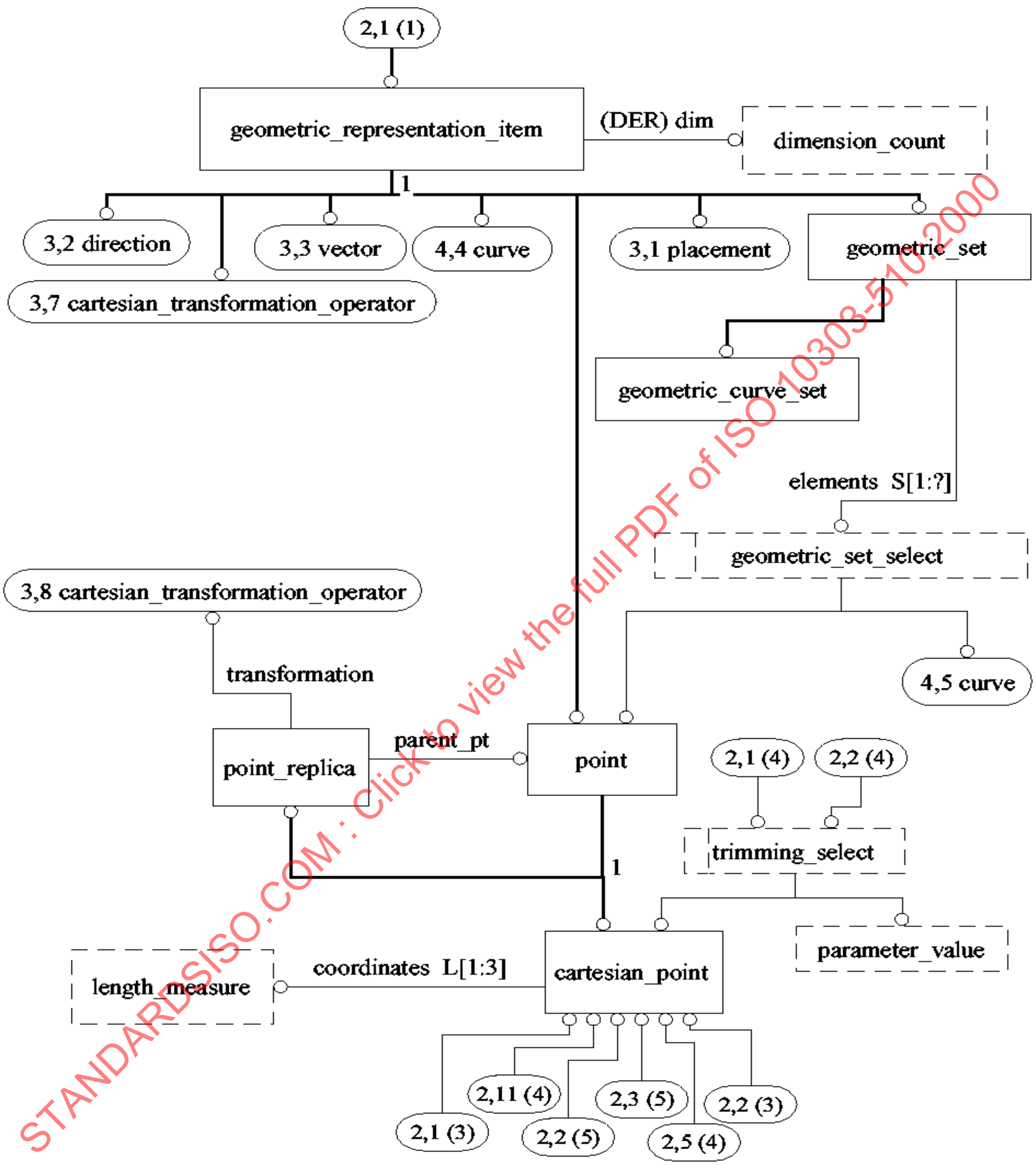

Figure C.2 - aic_geometrically_bounded_wireframe - EXPRESS-G diagram 2 of 6

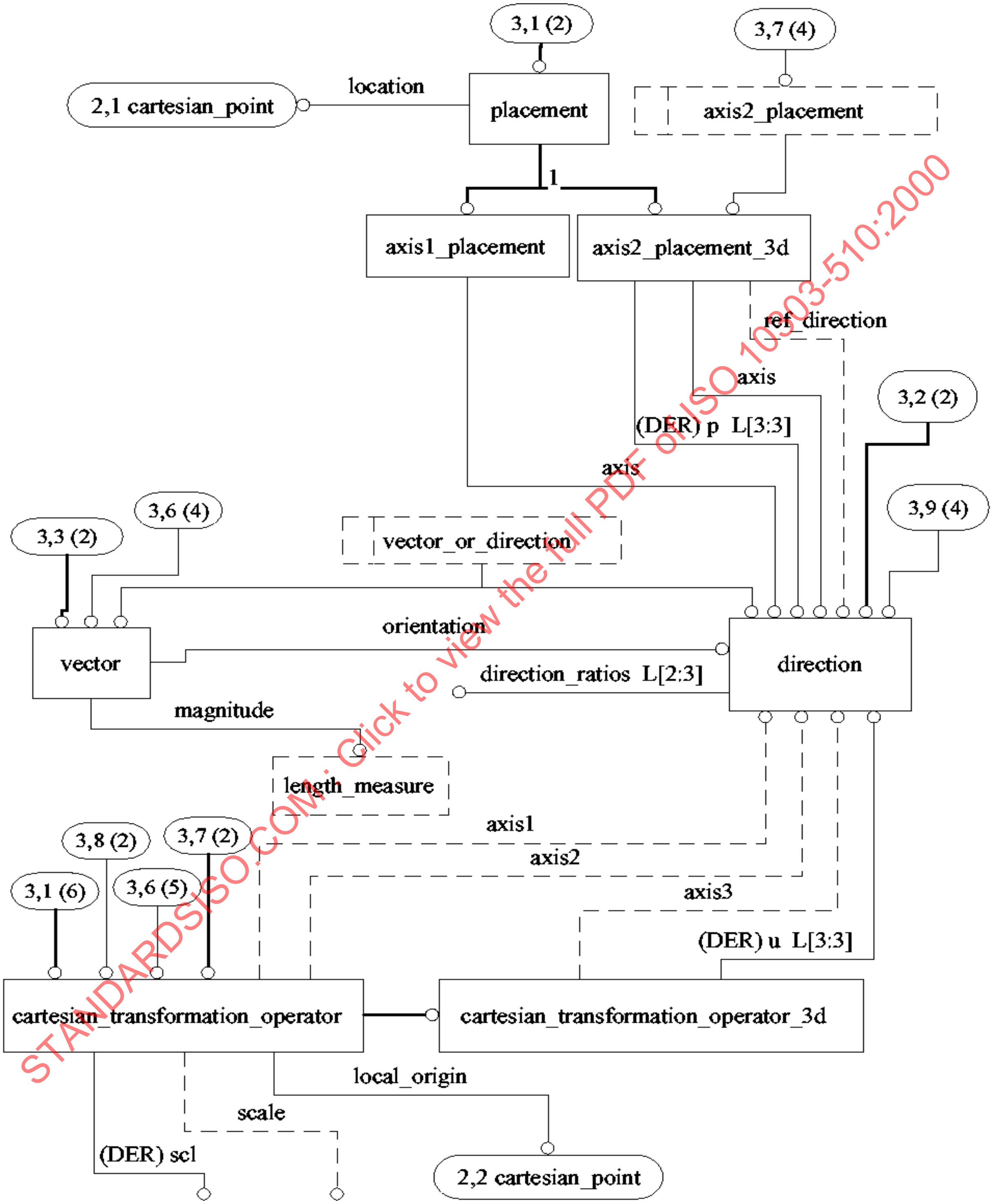

Figure C.3 - aic_geometrically_bounded_wireframe - EXPRESS-G diagram 3 of 6