

**Implementation of
ISO 10303-42 : 1994**

**Industrial automation systems
and integration —**

**Product data representation
and exchange**

**Part 42. Integrated generic resources:
Geometric and topological representation**



*
5
*

Committees responsible for this British Standard

The preparation of this British Standard was entrusted to Technical Committee AMT/4, Product data exchange, upon which the following bodies were represented:

- British Aerospace
- CIMIO Ltd.
- Construction Industry Computing Association
- Department of Trade and Industry (Services Management Division (IMS))
- GKN Technology Ltd.
- Institute of Materials
- Institution of Chemical Engineers
- Institution of Electrical Engineers
- Lloyd's Register of Shipping
- Ministry of Defence
- NAFEMS
- Nine Tiles K-Net
- Rutherford Appleton Laboratory
- South Bank University
- University of Bristol
- University of Leeds Industrial Services Ltd.
- University of Loughborough

This British Standard, having been prepared under the direction of the Advanced Manufacturing Technology Assembly, was published under the authority of the Standards Board and comes into effect on 15 October 1995

© BSI 1995

Amendments issued since publication

Amd. No.	Date	Text affected

The following BSI references relate to the work on this standard:
Committee reference AMT/4

Draft for comment 93/614095 DC03-42:1994

National foreword

This British Standard reproduces verbatim ISO 10303-42 : 1994 and implements it as the UK national standard.

This British Standard is published under the direction of the Advanced Manufacturing Technology Assembly whose Technical Committee AMT/4 has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international committee any enquiries on interpretation, or proposals for change, and keep UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

NOTE. International and European Standards, as well as overseas standards, are available from Customer Services, BSI, 389 Chiswick High Road, London W4 4AL.

Compliance with a British Standard does not of itself confer immunity from legal obligations.



* S *



INTERNATIONAL STANDARD

ISO
10303-42

First edition
1994-12-15

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

Part 42:

Integrated generic resources: Geometric and
topological representation

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

*Partie 42: Ressources génériques intégrées: Représentation géométrique
et topologique*



AVA Documents
Document Sharing Hub

Reference number
ISO 10303-42:1994(E)

Contents

	Page
1 Scope	1
1.1 Geometry	1
1.2 Topology	2
1.3 Geometric Shape Models	2
2 Normative references	3
3 Definitions, symbols and abbreviations	3
3.1 Definitions	3
3.1.1 arcwise connected	3
3.1.2 axi-symmetric	3
3.1.3 bounds	4
3.1.4 boundary	4
3.1.5 boundary representation solid model (Brep)	4
3.1.6 closed curve	4
3.1.7 closed surface	4
3.1.8 completion of a topological entity	4
3.1.9 connected	4
3.1.10 connected component	4
3.1.11 constructive solid geometry (CSG)	4
3.1.12 coordinate space	4
3.1.13 curve	4
3.1.14 cycle	4
3.1.15 d-manifold with boundary	5
3.1.16 dimensionality	5
3.1.17 domain	5
3.1.18 euler equations	5
3.1.19 extent	5
3.1.20 finite	5
3.1.21 genus of a graph	5
3.1.22 genus of a surface	5
3.1.23 geometrically founded	5
3.1.24 geometrically related	5

© ISO 1994

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Organization for Standardization
Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland



Documents

Document Sharing Hub

3.1.25	geometric coordinate system	5
3.1.26	graph	5
3.1.27	handle	6
3.1.28	homeomorphic	6
3.1.29	inside	6
3.1.30	interior	6
3.1.31	list	6
3.1.32	model space	6
3.1.33	open curve	6
3.1.34	open surface	6
3.1.35	orientable	6
3.1.36	overlap	6
3.1.37	parameter range	6
3.1.38	parameter space	6
3.1.39	placement coordinate system	6
3.1.40	self-intersect	7
3.1.41	self-loop	7
3.1.42	set	7
3.1.43	space dimensionality	7
3.1.44	surface	7
3.1.45	topological sense	7
3.2	Symbols	7
3.2.1	Geometry and mathematical symbology	7
3.2.2	Topology symbols	7
3.3	Abbreviations	10
4	Geometry	11
4.1	Introduction	12
4.2	Fundamental concepts and assumptions	12
4.2.1	Space dimensionality	12
4.2.2	Geometric relationships	12
4.2.3	Parametrisation of analytic curves and surfaces	13
4.2.4	Curves	13
4.2.5	Surfaces	13
4.2.6	Preferred form	14
4.3	geometry_schema type definitions	14
4.3.1	dimension_count	14
4.3.2	transition_code	14
4.3.3	preferred_surface_curve_representation	15
4.3.4	b_spline_curve_form	15
4.3.5	b_spline_surface_form	16
4.3.6	knot_type	17
4.3.7	extent_enumeration	18
4.3.8	trimming_preference	18
4.3.9	axis2_placement	19
4.3.10	curve_on_surface	19

ISO 10303-42:1994(E)

4.3.11	pcurve_or_surface	19
4.3.12	trimming_select	20
4.3.13	vector_or_direction	20
4.4	geometry_schema entity definitions	20
4.4.1	geometric_representation_context	20
4.4.2	geometric_representation_item	21
4.4.3	point	22
4.4.4	cartesian_point	23
4.4.5	point_on_curve	23
4.4.6	point_on_surface	24
4.4.7	point_replica	24
4.4.8	degenerate_pcurve	25
4.4.9	evaluated_degenerate_pcurve	26
4.4.10	direction	26
4.4.11	vector	27
4.4.12	placement	27
4.4.13	axis1_placement	28
4.4.14	axis2_placement_2d	28
4.4.15	axis2_placement_3d	29
4.4.16	cartesian_transformation_operator	31
4.4.17	cartesian_transformation_operator_3d	33
4.4.18	cartesian_transformation_operator_2d	36
4.4.19	curve	37
4.4.20	line	37
4.4.21	conic	38
4.4.22	circle	38
4.4.23	ellipse	39
4.4.24	hyperbola	41
4.4.25	parabola	43
4.4.26	bounded_curve	44
4.4.27	polyline	45
4.4.28	b_spline_curve	45
4.4.29	b_spline_curve_with_knots	48
4.4.30	uniform_curve	50
4.4.31	quasi_uniform_curve	50
4.4.32	bezier_curve	51
4.4.33	rational_b_spline_curve	52
4.4.34	trimmed_curve	54
4.4.35	composite_curve	56
4.4.36	composite_curve_segment	57
4.4.37	reparametrised_composite_curve_segment	59
4.4.38	pcurve	59
4.4.39	bounded_pcurve	60
4.4.40	surface_curve	61
4.4.41	intersection_curve	62
4.4.42	seam_curve	62

4.4.43	bounded_surface_curve	64
4.4.44	composite_curve_on_surface	64
4.4.45	offset_curve_2d	65
4.4.46	offset_curve_3d	66
4.4.47	curve_replica	67
4.4.48	surface	68
4.4.49	elementary_surface	69
4.4.50	plane	69
4.4.51	cylindrical_surface	70
4.4.52	conical_surface	71
4.4.53	spherical_surface	73
4.4.54	toroidal_surface	74
4.4.55	degenerate_toroidal_surface	75
4.4.56	swept_surface	76
4.4.57	surface_of_linear_extrusion	76
4.4.58	surface_of_revolution	77
4.4.59	bounded_surface	78
4.4.60	b_spline_surface	78
4.4.61	b_spline_surface_with_knots	81
4.4.62	uniform_surface	83
4.4.63	quasi_uniform_surface	83
4.4.64	bezier_surface	84
4.4.65	rational_b_spline_surface	85
4.4.66	rectangular_trimmed_surface	86
4.4.67	curve_bounded_surface	87
4.4.68	boundary_curve	89
4.4.69	outer_boundary_curve	89
4.4.70	rectangular_composite_surface	90
4.4.71	surface_patch	91
4.4.72	offset_surface	92
4.4.73	surface_replica	93
4.5	geometry_schema rule definitions	94
4.5.1	compatible_dimension	94
4.6	geometry_schema function definitions	95
4.6.1	dimension_of	95
4.6.2	acyclic_curve_replica	96
4.6.3	acyclic_point_replica	97
4.6.4	acyclic_surface_replica	97
4.6.5	associated_surface	98
4.6.6	base_axis	99
4.6.7	build_2axes	100
4.6.8	build_axes	100
4.6.9	orthogonal_complement	101
4.6.10	first_proj_axis	102
4.6.11	second_proj_axis	103
4.6.12	cross_product	103

4.6.13	dot_product	104
4.6.14	normalise	105
4.6.15	scalar_times_vector	107
4.6.16	vector_sum	108
4.6.17	vector_difference	109
4.6.18	default_b_spline_knot_mult	110
4.6.19	default_b_spline_knots	111
4.6.20	default_b_spline_curve_weights	112
4.6.21	default_b_spline_surface_weights	113
4.6.22	constraints_param_b_spline	114
4.6.23	curve_weights_positive	115
4.6.24	constraints_composite_curve_on_surface	116
4.6.25	get_basis_surface	116
4.6.26	surface_weights_positive	117
4.6.27	constraints_rectangular_composite_surface	118
4.6.28	list_to_array	119
4.6.29	make_array_of_array	120
5	Topology	122
5.1	Introduction	122
5.2	Fundamental concepts and assumptions	122
5.2.1	Geometric associations	123
5.2.2	Associations with parameter space geometry	124
5.2.3	Graphs, cycles, and traversals	126
5.3	topology_schema type definitions	127
5.3.1	shell	127
5.3.2	reversible_topology_item	128
5.3.3	list_of_reversible_topology_item	128
5.3.4	set_of_reversible_topology_item	128
5.3.5	reversible_topology	128
5.4	topology_schema entity definitions	129
5.4.1	topological_representation_item	129
5.4.2	vertex	129
5.4.3	vertex_point	130
5.4.4	edge	130
5.4.5	edge_curve	132
5.4.6	oriented_edge	133
5.4.7	path	133
5.4.8	oriented_path	134
5.4.9	open_path	135
5.4.10	loop	136
5.4.11	vertex_loop	137
5.4.12	edge_loop	137
5.4.13	poly_loop	138
5.4.14	face_bound	139
5.4.15	face_outer_bound	139

5.4.16	face	140
5.4.17	face_surface	142
5.4.18	oriented_face	143
5.4.19	subface	143
5.4.20	connected_face_set	144
5.4.21	vertex_shell	144
5.4.22	wire_shell	145
5.4.23	open_shell	146
5.4.24	oriented_open_shell	149
5.4.25	closed_shell	150
5.4.26	oriented_closed_shell	151
5.4.27	connected_edge_set	152
5.5	topology_schema function definitions	153
5.5.1	conditional_reverse	153
5.5.2	topology_reversed	153
5.5.3	edge_reversed	154
5.5.4	path_reversed	155
5.5.5	face_bound_reversed	155
5.5.6	face_reversed	156
5.5.7	shell_reversed	156
5.5.8	set_of_topology_reversed	157
5.5.9	list_of_topology_reversed	158
5.5.10	boolean_choose	158
5.5.11	path_head_to_tail	159
5.5.12	list_face_loops	159
5.5.13	list_loop_edges	160
5.5.14	list_shell_edges	160
5.5.15	list_shell_faces	161
5.5.16	list_shell_loops	161
5.5.17	mixed_loop_type_set	162
5.5.18	list_to_set	163
5.5.19	edge_curve_pcurves	163
5.5.20	vertex_point_pcurves	164
6	Geometric models	166
6.1	Introduction	166
6.2	Fundamental concepts and assumptions	166
6.3	geometric_model_schema type definitions	167
6.3.1	boolean_operand	167
6.3.2	boolean_operator	167
6.3.3	csg_primitive	168
6.3.4	csg_select	168
6.3.5	geometric_set_select	169
6.3.6	surface_model	169
6.3.7	wireframe_model	169
6.4	geometric_model_schema entity definitions	170

ISO 10303-42:1994(E)

6.4.1	solid_model	170
6.4.2	manifold_solid_brep	170
6.4.3	brep_with_voids	173
6.4.4	faceted_brep	173
6.4.5	csg_solid	174
6.4.6	boolean_result	175
6.4.7	sphere	175
6.4.8	right_circular_cone	176
6.4.9	right_circular_cylinder	177
6.4.10	torus	177
6.4.11	block	178
6.4.12	right_angular_wedge	179
6.4.13	swept_face_solid	180
6.4.14	extruded_face_solid	181
6.4.15	revolved_face_solid	181
6.4.16	swept_area_solid	183
6.4.17	extruded_area_solid	183
6.4.18	revolved_area_solid	184
6.4.19	half_space_solid	185
6.4.20	boxed_half_space	185
6.4.21	box_domain	186
6.4.22	solid_replica	186
6.4.23	shell_based_surface_model	187
6.4.24	face_based_surface_model	188
6.4.25	shell_based_wireframe_model	188
6.4.26	edge_based_wireframe_model	189
6.4.27	geometric_set	190
6.4.28	geometric_curve_set	190
6.4.29	geometric_set_replica	190
6.5	geometric_model_schema function definitions	191
6.5.1	acyclic_solid_replica	191
6.5.2	acyclic_set_replica	192
6.5.3	constraints_geometry_shell_based_surface_model	192
6.5.4	constraints_geometry_shell_based_wireframe_model	193
6.5.5	build_transformed_set	194

Annexes

A	Short names of entities	196
B	Information object registration	202
C	Computer-interpretable listings	203
D	EXPRESS-G diagrams	204
E	Bibliography	223



Index	224
-------------	-----

Figures

1	Axis2 placement 3D	31
2	(a) Cartesian transformation operator 3D	33
2	(b) Cartesian transformation operator 3D	34
2	(c) Cartesian transformation operator	35
3	Circle	39
4	Ellipse	40
5	Hyperbola	42
6	Parabola	44
7	B-spline curve	47
8	Composite curve	58
9	Conical surface	72
10	Curve bounded surface	88
11	Edge curve	131
12	Right angular wedge and its attributes	179
13	Revolved face solid	182
D.1	geometry_schema EXPRESS-G diagram 1 of 12 - Subtypes of geometric representation item.	205
D.2	geometry_schema EXPRESS-G diagram 2 of 12 - Placement and transformation subtypes	206
D.3	geometry_schema EXPRESS-G diagram 3 of 12- Subtypes of point	207
D.4	geometry_schema EXPRESS-G diagram 4 of 12 - Subtypes of curve	208
D.5	geometry_schema EXPRESS-G diagram 5 of 12 - Subtypes of surface	209
D.6	geometry_schema EXPRESS-G diagram 6 of 12 - Subtypes of conic	210
D.7	geometry_schema EXPRESS-G diagram 7 of 12 - Surface curve	211
D.8	geometry_schema EXPRESS-G diagram 8 of 12 - Bounded curves	212
D.9	geometry_schema EXPRESS-G diagram 9 of 12 - B_spline_curve	213
D.10	geometry_schema EXPRESS-G diagram 10 of 12 - Elementary surface	214
D.11	geometry_schema EXPRESS-G diagram 11 of 12 - Subtypes of bounded surface	215
D.12	geometry_schema EXPRESS-G diagram 12 of 12 - B_spline_surface	216

D.13 topology_schema EXPRESS-G diagram 1 of 3 - Subtypes of topological representation
 item 217

D.14 topology_schema EXPRESS-G diagram 2 of 3 218

D.15 topology_schema EXPRESS-G diagram 3 of 3 219

D.16 geometric_model_schema EXPRESS-G diagram 1 of 3 - Subtypes of solid model . . . 220

D.17 geometric_model_schema EXPRESS-G diagram 2 of 3 - CSG solid 221

D.18 geometric_model_schema EXPRESS-G diagram 3 of 3 - Models of external shape . . 222

Tables

1 Geometry mathematical symbology 8

2 Topology Symbol Definitions 9

A.1 Short names of entities 196

Foreword

The International Organization for Standardization (ISO) 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 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 10303-42 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data and global manufacturing programming languages*.

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 21, Implementation methods: Clear text encoding of the exchange structure;
- Part 22, Implementation methods: Standard data access interface specification;
- Part 31, Conformance testing methodology and framework: General concepts;
- Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- 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 resources: Materials;
- Part 46, Integrated generic resources: Visual presentation;
- Part 47, Integrated generic resources: Shape variation tolerances;
- Part 49, Integrated generic resources: Process structure and properties;

ISO 10303-42:1994(E)

- Part 101, Integrated application resources: Draughting;
- Part 104, Integrated application resources: Finite element analysis;
- Part 105, Integrated application resources: Kinematics;
- Part 201, Application protocol: Explicit draughting;
- Part 202, Application protocol: Associative draughting;
- Part 203, Application protocol: Configuration controlled design;
- Part 207, Application protocol: Sheet metal die planning and design;
- Part 210, Application protocol: Printed circuit assembly product design data;
- Part 213, Application protocol: Numerical control process plans for machined parts.

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

- Part 11 specifies the description methods;
- Parts 21 and 22 specify the implementation methods;
- Parts 31 and 32 specify the conformance testing methodology and framework;
- Parts 41 to 49 specify the integrated generic resources;
- Parts 101 to 105 specify the integrated application resources;
- Parts 201 to 213 specify the application protocols.

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. Annexes C, D, E are for information only.

Diskette

Users should note that this part of ISO 10303 comprises a diskette:

- the short names of entities given in annex A are also included on the diskette;
- the EXPRESS listings (annex C) are provided on the diskette only;
- a method to enable users to report errors in the documentation is given. Full details are provided in the file.

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 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 integrated resources series. Major subdivisions of this International Standard are:

- Geometry
- Topology
- Geometric models

This part of ISO 10303 specifies the integrated resources used for geometric and topological representation. Their primary application is for explicit representation of the shape or geometric form of a product model. The shape representation presented here has been designed to facilitate stable and efficient communication when mapped to a physical file.

The geometry in clause 4 is exclusively the geometry of parametric curves and surfaces. It includes the curve and surface entities and other entities, functions and data types necessary for their definition. A common scheme has been used for the definition of both two-dimensional and three-dimensional geometry. All geometry is defined in a coordinate system which is established as part of the context of the item which it represents. These concepts are fully defined in ISO 10303 Part 43.

The topology in clause 5 is concerned with connectivity relationships between objects rather than with the precise geometric form of objects. This clause contains the basic topological entities and specialised subtypes of these. In some cases the subtypes have geometric associations. Also included are functions, particularly constraint functions, and data types necessary for the definitions of the topological entities.

The geometric models in clause 6 provide basic resources for the communication of data describing the precise size and shape of three-dimensional solid objects. The geometric shape models provide a complete representation of the shape which in many cases includes both geometric and topological data. Included here are the two classical types of solid model, constructive solid geometry (CSG) and boundary representation (B-rep). Other entities, providing a rather less complete description of the geometry of a product, and with less consistency constraints, are also included.





Industrial automation systems and integration — Product data representation and exchange — Part 42 : Integrated generic resources: Geometric and topological representation

1 Scope

This part of ISO 10303 specifies the resource constructs for the explicit geometric and topological representation of the shape of a product. The scope is determined by the requirements for the explicit representation of an ideal product model; tolerances and implicit forms of representation in terms of features are out of scope. The geometry in clause 4 and the topology in clause 5 are available for use independently and are also extensively used by the various forms of geometric shape model in clause 6. In addition, this part of ISO 10303 specifies specialisations of the concepts of representation where the elements of representation are geometric.

1.1 Geometry

The following are within the scope of the geometry schema:

- definition of points, vectors, parametric curves and parametric surfaces;
- definition of transformation operators;
- points defined directly by their coordinate values or in terms of the parameters of an existing curve or surface;
- definition of conic curves and elementary surfaces;
- definition of curves defined on a parametric surface;
- definition of general parametric spline curves and surfaces;
- definition of point, curve and surface replicas;
- definition of offset curves and surfaces;
- definition of intersection curves.

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

- all other forms of procedurally defined curves and surfaces;



Documents

Document Sharing Hub

- curves and surfaces which do not have a parametric form of representation;
- any form of explicit representation of a ruled surface.

NOTE – For a ruled surface the geometry is critically dependent upon the parametrisation of the boundary curves and the method of associating pairs of points on the two curves. A ruled surface with B-spline boundary curves can however be exactly represented by the B-spline surface entity.

1.2 Topology

The following are within the scope of the topology schema:

- definition of the fundamental topological entities vertex, edge, and face, each with a specialised subtype to enable it to be associated with the geometry of a point, curve, or surface, respectively;
- collections of the basic entities to form topological structures of path, loop and shell and constraints to ensure the integrity of these structures;
- orientation of topological entities.

1.3 Geometric Shape Models

The following are within the scope of the geometric model schema:

- data describing the precise geometric form of three-dimensional solid objects;
- constructive solid geometry (CSG) models;
- definition of CSG primitives and half-spaces;
- creation of solid models by sweeping operations;
- manifold boundary representation (B-rep) models;
- constraints to ensure the integrity of B-rep models;
- surface models;
- wireframe models;
- geometric Sets;
- creation of a replica of a solid model in a new location.

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

- non-manifold boundary representation models;

- spatial occupancy forms of solid models (such as octree models);
- assemblies and mechanisms.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8824-1:-¹⁾, *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 EXPRESSLanguage 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 10303-43:1994, *Industrial automation systems and integration - Product data representation and exchange - Part 43: Integrated generic resources: Representation structures.*

3 Definitions, symbols and abbreviations

3.1 Definitions

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

3.1.1 arcwise connected: an entity is arcwise connected if any two arbitrary points in its domain can be connected by a curve that lies entirely within the domain.

3.1.2 axi-symmetric: an entity is axi-symmetric if it has an axis of symmetry such that the object is invariant under all rotations about this axis.

¹⁾To be published.

3.1.3 bounds: the topological entities of lower dimensionality which mark the limits of a topological entity. The bounds of a face are loops, and the bounds of an edge are vertices.

3.1.4 boundary: the set of mathematical points x in a domain X contained in R^m for which there is an open ball U in R^m containing x such that the intersection $U \cap X$ is homeomorphic to an open set in the closed d -dimensional half-space R_+^d , for some $d \leq m$, where the homeomorphism carries x into the origin in R_+^d .

NOTES

1 – R_+^d is defined to be the set of all mathematical points $(x_1, \dots, x_d)$ in R^d with $x_1 \geq 0$.

2 – For this purpose, the word “open” has its usual mathematical meaning. It does not relate to “open surface” as defined elsewhere in this part of ISO 10303.

3.1.5 boundary representation solid model (Brep): a type of geometric model in which the size and shape of the solid is defined in terms of the faces, edges and vertices which make up its boundary.

3.1.6 closed curve: a curve such that both end points are the same.

3.1.7 closed surface: a connected 2-manifold that divides space into exactly two connected components, one of which is finite.

3.1.8 completion of a topological entity: a set consisting of the entity in question together with all the faces, edges and vertices referenced, directly or indirectly, in the definition of the bounds of that entity.

3.1.9 connected: equivalent to arcwise connected.

3.1.10 connected component: a maximal connected subset of a domain.

3.1.11 constructive solid geometry (CSG): a type of geometric modelling in which a solid is defined as the result of a sequence of regularised Boolean operations operating on solid models.

3.1.12 coordinate space: a reference system that associates a unique set of n parameters with each point in an n -dimensional space.

3.1.13 curve: a set of mathematical points which is the image, in two- or three-dimensional space, of a continuous function defined over a connected subset of the real line (R^1), and which is not a single point.

3.1.14 cycle: a chain of alternating vertices and edges in a graph such that the first and last vertices are the same.

3.1.15 *d*-manifold with boundary: a domain which is the union of its *d*-dimensional interior and its boundary.

3.1.16 dimensionality: the number of independent coordinates in the parameter space of a geometric entity. The dimensionality of topological entities which need not have domains is specified in the entity definitions. The dimensionality of a list or set is the maximum of the dimensionalities of the elements of that list or set.

3.1.17 domain: the mathematical point set in model space corresponding to an entity.

3.1.18 euler equations: Equations used to verify the topological consistency of objects. Various equalities relating topological properties of entities are derived from the invariance of a number known as the Euler characteristic. Typically, these are used as quick checks on the integrity of the topological structure. A violation of an Euler condition signals an "impossible" object. Two special cases are important in this document. The Euler equation for graphs is discussed in 5.2.3. Euler conditions for surfaces are discussed in 5.4.23 and 5.4.25.

3.1.19 extent: the measure of the content of the domain of an entity, measured in units appropriate to the dimensionality of the entity. Thus, length, area and volume are used for dimensionalities 1, 2, and 3, respectively. Where necessary, the symbol Ξ will be used to denote extent.

3.1.20 finite: an entity is finite (sometimes called bounded) if there is a finite upper bound on the distance between any two points in its domain.

3.1.21 genus of a graph: the integer-valued invariant defined algorithmically by the graph traversal algorithm described in the note in 5.2.3.

3.1.22 genus of a surface: the number of handles that must be added to a sphere to produce a surface homeomorphic to the surface in question.

3.1.23 geometrically founded: a property of `geometric_representation_items` asserting their relationship to a coordinate space in which the coordinate values of points and directions on which they depend for position and orientation are measured.

3.1.24 geometrically related: the relationship between two `geometric_representation_items` in the same context by which the concepts of distance and direction between them are defined.

3.1.25 geometric coordinate system: the underlying global rectangular Cartesian coordinate system to which all geometry refers.

3.1.26 graph: a set of vertices and edges. The graphs discussed in this document are generally called pseudographs in the technical literature because they allow self-loops and also multiple edges connecting the same two vertices.

3.1.27 handle: the structure distinguishing a torus from a sphere, which can be viewed as a cylindrical tube connecting two holes in a surface.

3.1.28 homeomorphic: domains X and Y are homeomorphic if there is a continuous function f from X to Y which is a one-to-one correspondence, so that the inverse function f^{-1} exists, and if f^{-1} is also continuous.

3.1.29 inside: domain X is inside domain Y if both domains are contained in the same Euclidean space, R^m , and Y separates R^m into exactly two connected components, one of which is finite, and X is contained in the finite component.

3.1.30 interior: the d -dimensional interior of a d -dimensional domain X contained in R^m is the set of mathematical points x in X for which there is an open ball U in R^m containing x such that the intersection $U \cap X$ is homeomorphic to an open ball in R^d .

3.1.31 list: an ordered homogeneous collection with possibly duplicate members. A list is represented by an enclosing pair of brackets, i.e. $[A]$.

3.1.32 model space: a space with dimensionality 2 or 3 in which the geometry of a physical object is defined.

3.1.33 open curve: a curve which has two distinct end points.

3.1.34 open surface: a surface which is a manifold with boundary, but is not closed. I.e., either it is not finite, or it does not divide space into exactly two connected components.

3.1.35 orientable: a surface is orientable if a consistent, continuously varying choice can be made of the sense of the normal vectors to the surface.

NOTE – This does not require a continuously varying choice of the *values* of the normal vectors; the surface may have tangent plane discontinuities.

3.1.36 overlap: two entities overlap when they have shells, faces, edges, or vertices in common.

3.1.37 parameter range: the range of valid parameter values for a curve or surface.

3.1.38 parameter space: the one-dimensional space associated with a curve via its uniquely defined parametrisation or the two-dimensional space associated with a surface.

3.1.39 placement coordinate system: a rectangular Cartesian coordinate system associated with the placement of a geometric entity in space, used to describe the interpretation of the attributes and to associate a unique parametrisation with curve and surface entities.

3.1.40 self-intersect: a curve or surface self-intersects if there is a mathematical point in its domain which is the image of at least two points in the object's parameter range, and one of

those two points lies in the interior of the parameter range. A vertex, edge or face self-intersects if its domain does.

NOTE – A curve or surface is not considered to be self-intersecting just because it is closed.

3.1.41 self-loop: an edge that has the same vertex at both ends.

3.1.42 set: an unordered collection in which no two members are equal.

3.1.43 space dimensionality: the number of parameters required to define the location of a point in the coordinate space.

3.1.44 surface: a set of mathematical points which is the image of a continuous function defined over a connected subset of the plane (R^2).

3.1.45 topological sense: the sense of a topological entity as derived from the order of its attributes.

EXAMPLES

1 – The topological sense of an edge is from the edge start vertex to the edge end vertex.

2 – The topological sense of a path follows the edges in their listed order.

3.2 Symbols

For the purposes of this part of ISO 10303, the following symbols and definitions apply.

3.2.1 Geometry and mathematical symbology

The mathematical symbol convention used in the geometry schema is given in table 1.

3.2.2 Topology symbols

An attempt has been made to define precisely the constraints that shall be met by the topological entities. In many cases these are defined symbolically. This subclause describes the notation used for this purpose. It should be noted that the definitions given here are independent of EXPRESS definitions and usage.

The topological constructs are **vertex**, **edge**, **path**, **loop**, **face** (and **subface**) and **shell**. These will be referred to by the following symbols V , E , P , L , F and S , respectively.

Some of these entities take particular forms and a superscript is used to distinguish between these forms if necessary.

EXAMPLE 3 – A loop may be a **vertex loop**, an **edge loop** or a **poly loop**. These forms are denoted as L^v , L^e , L^p .

Table 1 – Geometry mathematical symbology

Symbol	Definition
a	Scalar quantity
A	Vector quantity
$\langle \rangle$	Vector normalisation
$\mathbf{a}$	Normalised vector (e.g. $\mathbf{a} = \langle A \rangle = A/ A $)
$\times$	Vector (cross) product
$\cdot$	Scalar product
$A \rightarrow B$	A is transformed to B
$\lambda(u)$	Parametric curve
$C(x, y, z)$	Analytic curve
$\sigma(u, v)$	Parametric surface
$S(x, y, z)$	Analytic surface
C_x	Partial differential of C with respect to x
σ_u	Partial derivative of $\sigma(u, v)$ with respect to u
S_x	Partial derivative of S with respect to x
$ $	Absolute value, or magnitude or determinant
R^m	m-dimensional real space

Table 2 lists the symbols used in the topology schema.

An undirected edge is an entity of type **edge** which is not of the subtype **oriented edge**. In some instances of the entity definitions, a topological attribute may take the form of a (topological + logical) pair, this is generally represented by the oriented subtype. A subscript is used to distinguish between the topological and the (topological + logical) pairing. For example, E and E_i or S^o and S_i^o .

Several topological entities use an Orientation Flag to indicate whether the direction of a referenced entity agrees with or is opposed to the direction of the referencing entity. If the Flag is TRUE, the direction of the referenced entity is correct but if the Flag is FALSE, the direction of the referenced entity should be (conceptually) reversed. It can happen that there are several Orientation Flags in the chain of entities from the high-level referencing entity to the low-level referenced entity. The direction of a low-level entity with respect to a high-level entity is obtained by evaluating the *not exclusive or* ($\odot$) of the chain of Orientation Flags. For example, a Face references a Loop + Loopflag, a Loop references an Edge + Edgeflag and an Edge references a Curve + Curveflag. The Face's "FaceCurveflag" is given by

$$\text{FaceCurveflag} = \text{Loopflag} \odot \text{Edgeflag} \odot \text{Curveflag}$$

where *not exclusive or* is interpreted as true if the two flags have the same value and is defined by the truth table:

$$\begin{aligned} T \odot T &= T \\ T \odot F &= F = F \odot T \\ F \odot F &= T. \end{aligned}$$

Table 2 – Topology Symbol Definitions

Symbol	Definition
V	Vertex
$\mathcal{V}$	Number of unique vertices
E	Undirected edge
$\mathcal{E}$	Number of unique undirected edges
E_l	Oriented edge
$\mathcal{E}_l$	Number of unique oriented edges
G^e	Edge genus
P	Path
$\mathcal{P}$	Number of unique paths
G^p	Path genus
L	Loop
$\mathcal{L}$	Number of unique loops
L_l	Face bound
$\mathcal{L}_l$	Number of unique face bounds
L^e	Edge loop
L^p	Poly loop
L^v	Vertex loop
G^l	Loop genus
F	Face
$\mathcal{F}$	Number of unique faces
H^f	Face genus
S	Shell
$\mathcal{S}$	Number of unique shells
S^c	Closed shell
S^o	Open shell
S^v	Vertex shell
S^w	Wire shell
H^s	Shell genus
Ξ	Extent
$\{A\}$	Set of entities of type A
$[A]$	List of entities of type A

Thus

$$F \odot T \odot F = T.$$

3.3 Abbreviations

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

B-rep: boundary representation solid model;

CSG: constructive solid geometry.

4 Geometry

The following EXPRESS declaration begins the `geometry_schema` and identifies the necessary external references.

EXPRESS specification:

```

*)
SCHEMA geometry_schema;
  REFERENCE FROM representation_schema
    (representation,
     functionally_defined_transformation,
     representation_item,
     representation_context,
     definitional_representation,
     item_in_context,
     using_representations);
  REFERENCE FROM measure_schema
    (length_measure,
     positive_length_measure,
     plane_angle_measure,
     plane_angle_unit,
     positive_plane_angle_measure,
     parameter_value,
     global_unit_assigned_context);
  REFERENCE FROM topology_schema
    (edge_curve,
     face_surface,
     poly_loop,
     vertex_point);
  REFERENCE FROM geometric_model_schema
    (solid_model,
     boolean_result,
     sphere,
     right_circular_cone,
     right_circular_cylinder,
     torus,
     block,
     right_angular_wedge,
     half_space_solid,
     shell_based_surface_model,
     face_based_surface_model,
     shell_based_wireframe_model,
     edge_based_wireframe_model,
     geometric_set);
(*

```

NOTES

- 1 - The schemas referenced above can be found in:
- | | |
|-------------------------------------|------------------------------------|
| <code>representation_schema</code> | ISO 10303-43 |
| <code>measure_schema</code> | ISO 10303-41 |
| <code>topology_schema</code> | clause 5 of this part of ISO 10303 |
| <code>geometric_model_schema</code> | clause 6 of this part of ISO 10303 |

2 - The references to `topology_schema` and to `geometric_model_schema` are required only for the definition of the `geometric_representation_item` supertype.

3 - See annex D, figures D.1 to D.12, for a graphical presentation of this schema.

4.1 Introduction

The subject of the `geometry_schema` is the geometry of parametric curves and surfaces. The `representation_schema` (see ISO 10303-43) and the `geometric_representation_context` defined in this Part of ISO 10303, provide the context in which the geometry is defined. This enables a distinction to be made between those items which are geometrically related and those existing in independent coordinate spaces. In particular, each `geometric_representation_item` has a `geometric_representation_context` which includes as an attribute the Euclidean dimension of its coordinate space. The coordinate system for this space is referred to as the geometric coordinate system in this clause. Units associated with `length_measures` and `plane_angle_measures` are assumed to be assigned globally within this context. A global rule (`compatible_dimension`) ensures that all `geometric_representation_items` in the same `geometric_representation_context` have the same space dimensionality. The space dimensionality `dim` is an inherited derived attribute of all subtypes of `geometric_representation_item`.

4.2 Fundamental concepts and assumptions

4.2.1 Space dimensionality

All geometry shall be defined in a right-handed rectangular Cartesian coordinate system with the same units on each axis. A common scheme has been used for the definition of both two-dimensional and three-dimensional geometry. Points and directions exist in both a two-dimensional and a three-dimensional form; these forms are distinguished solely by the presence, or absence, of a third coordinate value. Complex geometric entities are all defined using points and directions from which their space dimensionality can be deduced.

4.2.2 Geometric relationships

All `geometric_representation_items` which are included as items in a `representation` having a `geometric_representation_context` are geometrically related. Any such `geometric_representation_item` is said to be geometrically founded in the context of that `representation`. No geometric relationship, such as distance between points, is assumed to exist for `geometric_representation_items` occurring as items in different `representations`.

4.2.3 Parametrisation of analytic curves and surfaces

Each curve or surface specified here has a defined parametrisation. In some instances the definitions are in parametric terms. In others, the conic curves and elementary surfaces, the definitions are in geometric terms.

In this latter case a placement coordinate system is used to define the parametrisation. The geometric definitions contain some, but not all, of the data required for this. The relevant data to define this placement coordinate system is contained in the **axis2_placement** associated with the individual curve and surface entities.

4.2.4 Curves

The curve entities defined in 4.4 include lines, elementary conics, a general parametric polynomial curve, and some referentially or procedurally defined curves. All the curves have a well defined parametrisation which makes it possible to trim a curve or identify points on the curve by parameter value. For the conic curves, a method of representation is used which separates their geometric form from their orientation and position in space. In each case, the position and orientation information is conveyed by an **axis2_placement**. The general purpose parametric curve is represented by the **b_spline_curve** entity. This was selected as the most stable form of representation for the communication of all types of polynomial and rational parametric curves. With appropriate attribute values and subtypes, a **b_spline_curve** entity is capable of representing single span or spline curves of explicit polynomial, rational, Bézier or B-spline type. A **composite_curve** entity, which includes the facility to communicate continuity information at the curve-to-curve transition points, is provided for the construction of more complex curves.

The **offset_curve** and **curve_on_surface** types are curves defined with reference to other geometry. Separate **offset_curve** entities exist for 2D and 3D applications. The curve on surface entities include an **intersection_curve** which represents the intersection of two surfaces. Such a curve may be represented in 3D space or in the 2D parameter space of either of the surfaces.

4.2.5 Surfaces

The surface entities support the requirements of simple boundary representation (B-rep) solid modelling system and enable the communication of general polynomial and rational parametric surfaces. The simple surfaces are the planar, spherical, cylindrical, conical and toroidal surfaces, a **surface_of_revolution** and a **surface_of_linear_extrusion**. As with curves, all surfaces have an associated standard parametrisation. In many cases the surfaces, as defined, are unbounded; it is assumed that they will be bounded either explicitly or implicitly. Explicit bounding is achieved with the **rectangular_trimmed_surface** or **curve_bounded_surface** entities; implicit bounding requires the association of additional topological information to define a face.

The **b_spline_surface** entity and its subtypes provide the most general capability for the communication of all types of polynomial and rational biparametric surfaces. This entity uses control points as the most stable form of representation for the surface geometry. The **offset_surface** entity is intended for the communication of a surface obtained as a simple normal offset from

a given surface. capability to connect together a rectangular mesh of distinct surface patches, specifying the degree of continuity from patch to patch.

4.2.6 Preferred form

Some of the geometric entities provide the potential capability of defining an item of geometry in more than one way. Such multiple representations are accommodated by requiring the nomination of a 'preferred form' or 'master representation'. This is the form which is used to determine the parametrisation.

NOTE – The `master_representation` attribute acknowledges the impracticality of ensuring that multiple forms are indeed identical and allows the indication of a preferred form. This would probably be determined by the creator of the data. All characteristics, such as parametrisation, domain, and results of evaluation, for an entity having multiple representations, are derived from the master representation. Any use of the other representations is a compromise for practical considerations.

4.3 geometry_schema type definitions

4.3.1 dimension_count

A `dimension_count` is a positive integer used to define the coordinate space dimensionality of a `geometric_representation_context`.

EXPRESS specification:

```
*)  
TYPE dimension_count = INTEGER;  
WHERE  
    WR1: SELF > 0;  
END_TYPE;  
(*
```

Formal propositions:

WR1: A `dimension_count` shall be positive.

4.3.2 transition_code

This type conveys the continuity properties of a composite curve or surface. The continuity referred to is geometric, not parametric continuity.

EXPRESS specification:

```
*)  
TYPE transition_code = ENUMERATION OF  
    (discontinuous,  
     continuous,  
     cont_same_gradient,  
     cont_same_gradient_same_curvature);  
END_TYPE;  
(*
```

Enumerated item definitions:

discontinuous: The segments, or patches, do not join. This is permitted only at the boundary of the curve or surface to indicate that it is not closed.

continuous: The segments, or patches, join, but no condition on their tangents is implied.

cont_same_gradient: The segments, or patches, join, and their tangent vectors, or tangent planes, are parallel and have the same direction at the joint; equality of derivatives is not required.

cont_same_gradient_same_curvature: For a curve, the segments join, their tangent vectors are parallel and in the same direction, and their curvatures are equal at the joint; equality of derivatives is not required. For a surface this implies that the principal curvatures are the same and that the principal directions are coincident along the common boundary.

4.3.3 preferred_surface_curve_representation

This type is used to indicate the preferred form of representation for a surface curve, which is either a curve in geometric space or in the parametric space of the underlying surfaces.

EXPRESS specification:

```

*)
TYPE preferred_surface_curve_representation = ENUMERATION OF
  (curve_3d,
   pcurve_s1,
   pcurve_s2);
END_TYPE;
(*)

```

Enumerated item definitions:

curve_3d: The curve in three-dimensional space is preferred.

pcurve_s1: The first pcurve is preferred.

pcurve_s2: The second pcurve is preferred.

4.3.4 b_spline_curve_form

This type is used to indicate a particular geometric form represented by the B-spline curve.

EXPRESS specification:

```

*)
TYPE b_spline_curve_form = ENUMERATION OF
  (polyline_form,
   circular_arc,
   elliptic_arc,

```

```

    parabolic_arc,
    hyperbolic_arc,
    unspecified);
END_TYPE;
(*)

```

Enumerated item definitions:

polyline_form: A connected sequence of line segments represented by degree 1 B-spline basis functions.

circular_arc: An arc of a circle, or a complete circle represented by a B-spline curve.

elliptic_arc: An arc of an ellipse, or a complete ellipse, represented by a B-spline curve.

parabolic_arc: An arc of finite length of a parabola represented by a B-spline curve.

hyperbolic_arc: An arc of finite length of one branch of a hyperbola represented by a B-spline curve.

unspecified: A B-spline curve for which no particular form is specified.

4.3.5 b_spline_surface_form

This type is used to indicate that the B-spline surface represents a part of a surface of some specific form.

EXPRESS specification:

```

*)
TYPE b_spline_surface_form = ENUMERATION OF
    (plane_surf,
     cylindrical_surf,
     conical_surf,
     spherical_surf,
     toroidal_surf,
     surf_of_revolution,
     ruled_surf,
     generalised_cone,
     quadric_surf,
     surf_of_linear_extrusion,
     unspecified);
END_TYPE;
(*)

```

Enumerated item definitions:

plane_surf: A bounded portion of a plane represented by a B-spline surface of degree 1 in each parameter.

cylindrical_surf: A bounded portion of a cylindrical surface.

conical_surf: A bounded portion of the surface of a right circular cone.

spherical_surf: A bounded portion of a sphere, or a complete sphere, represented by a B-spline surface.

toroidal_surf: A torus, or portion of a torus, represented by a B-spline surface.

surf_of_revolution: A bounded portion of a surface of revolution.

ruled_surf: A surface constructed from two parametric curves by joining with straight lines corresponding points with the same parameter value on each of the curves.

generalised_cone: A special case of a ruled surface in which the second curve degenerates to a single point; when represented by a B-spline surface all the control points along one edge will be coincident.

quadric_surf: A bounded portion of one of the class of surfaces of degree 2 in the variables x , y and z .

surf_of_linear_extrusion: A bounded portion of a surface of linear extrusion represented by a B-spline surface of degree 1 in one of the parameters.

unspecified: A surface for which no particular form is specified.

4.3.6 knot_type

This type indicates that the B-spline knots shall have a particularly simple form enabling the knots themselves to be defaulted.

For details of the interpretation of these types see the B-spline curve entity definition (4.4.29).

EXPRESS specification:

```
*)
TYPE knot_type = ENUMERATION OF
  (uniform_knots,
   unspecified,
   quasi_uniform_knots,
   piecewise_bezier_knots);
END_TYPE;
(*
```

Enumerated item definitions:

uniform_knots: The form of knots appropriate for a uniform B-spline curve.

unspecified: The type of knots is not specified. This includes the case of non uniform knots.

quasi_uniform_knots: The form of knots appropriate for a quasi-uniform B-spline curve.

piecewise_bezier_knots: The form of knots appropriate for a piecewise Bézier curve.

4.3.7 extent Enumeration

This type is used to describe the quantitative extent of an object.

EXPRESS specification:

```
*)
TYPE extent Enumeration = ENUMERATION OF
  (invalid,
   zero,
   finite_non_zero,
   infinite);
END_TYPE;
(*
```

Enumerated item definitions:

invalid: The concept of extent is not valid for the quantity being measured.

zero: The extent is zero.

finite_non_zero: The extent is finite (bounded) but not zero.

infinite: The extent is not finite.

4.3.8 trimming_preference

This type is used to indicate the preferred way of trimming a parametric curve where the trimming is multiply defined.

EXPRESS specification:

```
*)
TYPE trimming_preference = ENUMERATION OF
  (cartesian, parameter,
   unspecified);
END_TYPE;
(*
```

Enumerated item definitions:

cartesian: Indicates that trimming by cartesian point is preferred.

parameter: Indicates a preference for the parameter value.

unspecified: Indicates that no preference is communicated.

4.3.9 axis2_placement

This select type collects together both versions of the axis2 placement as used in two-dimensional or in three-dimensional Cartesian space. This enables entities requiring this information to reference them without specifying the space dimensionality.

EXPRESS specification:

```
*)
TYPE axis2_placement = SELECT
  (axis2_placement_2d,
   axis2_placement_3d);
END_TYPE;
(*
```

4.3.10 curve_on_surface

A **curve_on_surface** is a curve on a parametric surface. It may be any of the following

- a **pcurve** or
- a **surface_curve**, including the specialised subtypes of **intersection_curve** and **seam_curve**, or
- a **composite_curve_on_surface**.

The **curve_on_surface** select type collects these curves together for reference purposes.

EXPRESS specification:

```
*)
TYPE curve_on_surface = SELECT
  (pcurve,
   surface_curve,
   composite_curve_on_surface);
END_TYPE;
(*
```

4.3.11 pcurve_or_surface

This select type enables a surface curve to identify as an attribute the associated surface or pcurve.

EXPRESS specification:

```
*)
TYPE pcurve_or_surface = SELECT
  (pcurve,
   surface);
END_TYPE;
(*
```

4.3.12 **trimming_select**

This select type identifies the two possible ways of trimming a parametric curve, by a cartesian point on the curve, or by a REAL number defining a parameter value within the parametric range of the curve.

EXPRESS specification:

```
*)
TYPE trimming_select = SELECT
  (cartesian_point,
   parameter_value);
END_TYPE;
(*
```

4.3.13 **vector_or_direction**

This type is used to identify the types of entity which can participate in vector computations.

EXPRESS specification:

```
*)
TYPE vector_or_direction = SELECT
  (vector,
   direction);
END_TYPE;
(*
```

4.4 **geometry_schema entity definitions**

This subclause contains all the explicit geometric entities. Except for entities defined in a parameter space, all geometry is defined in a right-handed cartesian coordinate system (the geometric coordinate system). The space dimensionality of this coordinate system is established by the context of the **geometric_representation_item**. The curve and surface definitions are all given essentially in terms of points and/or vectors and/or scalar (length) values.

4.4.1 **geometric_representation_context**

A **geometric_representation_context** is a **representation_context** in which **geometric_representation_items** are geometrically founded.

A **geometric_representation_context** is a distinct coordinate space, spatially unrelated to other coordinate spaces except as those coordinate spaces are specifically related by an appropriate transformation. (See 3.1 for definitions of geometrically founded and coordinate space.)

EXPRESS specification:

```

*)
ENTITY geometric_representation_context
  SUBTYPE OF (representation_context);
  coordinate_space_dimension : dimension_count;
END_ENTITY;
(*)

```

Attribute definitions:

coordinate_space_dimension: The integer **dimension_count** of the coordinate space which is the **geometric_representation_context**.

NOTE - Any constraints on the allowed range of **coordinate_space_dimension** are outside the scope of this part of ISO 10303.

4.4.2 geometric_representation_item

A **geometric_representation_item** is a **representation_item** that has the additional meaning of having geometric position or orientation or both. This meaning is present by virtue of:

- being a **cartesian_point** or a **direction**;
- referencing directly a **cartesian_point** or a **direction**;
- referencing indirectly a **cartesian_point** or a **direction**.

NOTE 1 - An indirect reference to a **cartesian_point** or **direction** means that a given **geometric_representation_item** references the **cartesian_point** or **direction** through one or more intervening attributes. In many cases this information is given in the form of an **axis2_placement**.

EXAMPLES

4 - Consider a circle. It gains its geometric position and orientation by virtue of a reference to **axis2_placement** that in turn references a **cartesian_point** and several **directions**.

5 - A **manifold_solid_brep** is a **geometric_representation_item** that through several layers of **topological_representation_items**, references **curves**, **surfaces** and **points**. Through additional intervening entities **curves** and **surfaces** reference **cartesian_point** and **direction**.

NOTES

2 - The intervening entities, which are all of type **representation_item**, need not be of subtype **geometric_representation_item**. Consider the **manifold_solid_brep** from the above example. One of the intervening levels of **representation_item** is a **closed_shell**. This is a **topological_representation_item** and does not require a **geometric_representation_context** in its own right. When used as part of the definition of a **manifold_solid_brep** that itself is a **geometric_representation_item**, it is founded in a **geometric_representation_context**.

3 - A **geometric_representation_item** inherits the need to be related to a **representation_context** in a **representation**. The rule **compatible_dimension** ensures that the **representation_context** is a **geometric_representation_context**. When in the context of geometry, this relationship causes the **geometric_representation_item** to be geometrically founded.

EXPRESS specification:

```

*)
ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF(point, direction, vector, placement,
    cartesian_transformation_operator, curve, surface,
    edge_curve, face_surface, poly_loop, vertex_point,
    solid_model, boolean_result, sphere, right_circular_cone,
    right_circular_cylinder, torus, block,
    right_angular_wedge, half_space_solid,
    shell_based_surface_model, face_based_surface_model,
    shell_based_wireframe_model, edge_based_wireframe_model,
    geometric_set))
  SUBTYPE OF (representation_item);
DERIVE
  dim : dimension_count := dimension_of(SELF);
WHERE
  WR1: SIZEOF (QUERY (using_rep <= using_representations (SELF) |
    NOT ('GEOMETRY_SCHEMA.GEOMETRIC_REPRESENTATION_CONTEXT' IN
    TYPEOF (using_rep.context_of_items)))) = 0;
END_ENTITY;
(*

```

Attribute definitions:

dim: The coordinate dimension_count of the geometric_representation_item.

NOTES

4 - The dim attribute is derived from the coordinate_space_dimension of a geometric_representation_context in which the geometric_representation_item is geometrically founded.

5 - A geometric_representation_item is geometrically founded in one or more geometric_representation_contexts, all of which have the same coordinate_space_dimension. See the rule compatible_dimension in 4.5.1.

Formal propositions:

WR1: The context of any representation referencing a geometric_representation_item shall be of the type geometric_representation_context.

4.4.3 point

A point is a location in some real Cartesian coordinate space R^m , for $m = 1, 2$ or 3 .

EXPRESS specification:

```

*)
ENTITY point
  SUPERTYPE OF (ONEOF(cartesian_point, point_on_curve, point_on_surface,
    point_replica, degenerate_pcurve))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY;
(*

```

4.4.4 cartesian_point

A **cartesian_point** is a point defined by its coordinates in a rectangular Cartesian coordinate system, or in a parameter space. The entity is defined in a one, two or three-dimensional space as determined by the number of coordinates in the list.

NOTE – For the purposes of defining geometry in this part of ISO 10303 only two or three-dimensional points are used.

EXPRESS specification:

```
*)
ENTITY cartesian_point
  SUBTYPE OF (point);
  coordinates : LIST [1:3] OF length_measure;
END_ENTITY;
(*
```

Attribute definitions:

coordinates[1]: The first coordinate of the point location.

coordinates[2]: The second coordinate of the point location; this will not exist in the case of a one-dimensional point.

coordinates[3]: The third coordinate of the point location; this will not exist in the case of a one or two-dimensional point.

SELF\geometric_representation_item.dim: The dimensionality of the space in which the point is defined. This is an inherited derived attribute from the geometric representation item supertype and for a cartesian point is determined by the number of coordinates in the list.

4.4.5 point_on_curve

A **point_on_curve** is a point which lies on a curve. The point is determined by evaluating the curve at a specific parameter value. The coordinate space dimensionality of the point is that of the **basis_curve**.

EXPRESS specification:

```
*)
ENTITY point_on_curve
  SUBTYPE OF (point);
  basis_curve : curve;
  point_parameter : parameter_value;
END_ENTITY;
(*
```

Attribute definitions:

basis_curve: The curve to which **point_parameter** relates.

point_parameter: The parameter value of the point location.

SELF\geometric_representation_item.dim: The dimensionality of the space in which the **point_on_curve** is defined. This is the same as that of the **basis_curve**.

Informal propositions:

IP1: The value of the **point_parameter** shall not be outside the parametric range of the curve.

4.4.6 point_on_surface

A **point_on_surface** is a point which lies on a parametric surface. The point is determined by evaluating the surface at a particular pair of parameter values.

EXPRESS specification:

```
*)
ENTITY point_on_surface
  SUBTYPE OF (point);
  basis_surface      : surface;
  point_parameter_u  : parameter_value;
  point_parameter_v  : parameter_value;
END_ENTITY;
(*
```

Attribute definitions:

basis_surface: The surface to which the parameter values relate.

point_parameter_u: The first parameter value of the **point** location.

point_parameter_v: The second parameter value of the **point** location.

SELF\geometric_representation_item.dim: The dimensionality of the coordinate space of the **point_on_surface**. This is the same as that of the **basis_surface**.

Informal propositions:

IP1: The parametric values specified for **u** and **v** shall not be outside the parametric range of the **basis_surface**.

4.4.7 point_replica

This defines a replica of an existing point (the parent) in a different location. The replica has the same coordinate space dimensionality as the parent point.

EXPRESS specification:

```
*)
ENTITY point_replica
  SUBTYPE OF (point);
  parent_pt          : point;
  transformation      : cartesian_transformation_operator;
WHERE
  WR1: transformation.dim = parent_pt.dim;
```

```

    WR2: acyclic_point_replica (SELF,parent_pt);
END_ENTITY;
(*)

```

Attribute definitions:

parent_pt: The point to be replicated.

transformation: The Cartesian transformation operator which defines the location of the point replica.

Formal propositions:

WR1: The coordinate space dimensionality of the transformation attribute shall be the same as that of the **parent_pt**.

WR2: A **point_replica** shall not participate in its own definition.

4.4.8 degenerate_pcurve

A **degenerate_pcurve** is an entity with the structure of a **pcurve**, but which in three-dimensional model space collapses to a single point. It is thus a subtype of point, not of curve.

NOTE – For example, the apex of a cone could be represented as a degenerate pcurve.

EXPRESS specification:

```

*)
ENTITY degenerate_pcurve
  SUBTYPE OF (point);
  basis_surface: surface;
  reference_to_curve : definitional_representation;
WHERE
  WR1: SIZEOF(reference_to_curve\representation.items) = 1;
  WR2: 'GEOMETRY_SCHEMA.CURVE' IN TYPEOF
      (reference_to_curve\representation.items[1]);
  WR3: reference_to_curve\representation.
      items[1]\geometric_representation_item.dim =2;
END_ENTITY;
(*)

```

Attribute definitions:

basis_surface: The surface on which the **basis_curve** lies.

reference_to_curve: The association of the **pcurve** and the parameter space curve which degenerates to the (equivalent) point.

Formal propositions:

WR1: The set of items in the **definitional_representation** entity corresponding to the **reference_to_curve** shall have exactly one element.

WR2: The unique item in the set shall be a curve.

WR3: The dimensionality of this parameter space curve shall be 2.

Informal propositions:

IP1: Regarded as a curve in model space, the **degenerate_pcurve** shall have zero arc length.

4.4.9 evaluated_degenerate_pcurve

This entity represents the result of evaluating a degenerate pcurve and associates it with a Cartesian point.

EXPRESS specification:

```
*)
ENTITY evaluated_degenerate_pcurve
  SUBTYPE OF (degenerate_pcurve);
  equivalent_point : cartesian_point;
END_ENTITY;
(*
```

Attribute definitions:

equivalent_point: The point in the geometric coordinate system represented by the degenerate pcurve.

4.4.10 direction

This entity defines a general direction vector in two or three dimensional space. The actual magnitudes of the components have no effect upon the direction being defined, only the ratios x:y:z or x:y are significant.

NOTE – The components of this entity are not normalised. If a unit vector is required it should be normalised before use.

EXPRESS specification:

```
*)
ENTITY direction
  SUBTYPE OF (geometric_representation_item);
  direction_ratios : LIST [2:3] OF REAL;
WHERE
  WR1: SIZEOF(QUERY(tmp <= direction_ratios | tmp <> 0.0)) > 0;
END_ENTITY;
(*
```

Attribute definitions:

direction_ratios[1]: The component in the direction of the X axis.

direction_ratios[2]: The component in the direction of the Y axis.

direction_ratios[3]: The component in the direction of the Z axis; this will not be present in the case of a direction in two-dimensional coordinate space.

SELF\geometric_representation_item.dim: The coordinate space dimensionality of the direction. This is an inherited attribute of the **geometric_representation_item** supertype; for this entity it is determined by the number of **direction_ratios** in the list.

Formal propositions:

WR1: The magnitude of the direction vector shall be greater than zero.

4.4.11 vector

This entity defines a vector in terms of the direction and the magnitude of the vector. The value of the **magnitude** attribute defines the magnitude of the vector.

NOTE – The magnitude of the vector must not be calculated from the components of the **orientation** attribute. This form of representation was selected to reduce problems with numerical instability. For example a vector of magnitude 2.0 mm and equally inclined to the coordinate axes could be represented with orientation attribute of (1.0,1.0,1.0).

EXPRESS specification:

```
*)
ENTITY vector
  SUBTYPE OF (geometric_representation_item);
  orientation : direction;
  magnitude   : length_measure;
WHERE
  WR1 : magnitude >= 0.0;
END_ENTITY;
(*
```

Attribute definitions:

orientation: The direction of the vector.

magnitude: The magnitude of the vector. All vectors of **magnitude** 0.0 are regarded as equal in value regardless of the **orientation** attribute.

SELF\geometric_representation_item.dim: The dimensionality of the space in which the vector is defined.

Formal propositions:

WR1: The magnitude shall be positive or zero.

4.4.12 placement

A **placement** locates a geometric item with respect to the coordinate system of its geometric context. It locates the item to be defined and, in the case of the axis placement subtypes, gives its orientation.

EXPRESS specification:

```
*)
ENTITY placement
```

```

SUPERTYPE OF (ONEOF(axis1_placement,axis2_placement_2d,axis2_placement_3d))
SUBTYPE OF (geometric_representation_item);
location : cartesian_point;
END_ENTITY;
(*)

```

Attribute definitions:

location: The geometric position of a reference point, such as the centre of a circle, of the item to be located.

4.4.13 axis1_placement

The direction and location in three-dimensional space of a single axis. An **axis1_placement** is defined in terms of a locating point (inherited from the placement supertype) and an axis direction; this is either the direction of axis or defaults to (0.0,0.0,1.0). The actual direction for the axis placement is given by the derived attribute **z**.

EXPRESS specification:

```

*)
ENTITY axis1_placement
  SUBTYPE OF (placement);
  axis      : OPTIONAL direction;
DERIVE
  z          : direction := NVL(normalise(axis), direction([0.0,0.0,1.0]));
WHERE
  WR1: SELF\geometric_representation_item.dim = 3;
END_ENTITY;
(*)

```

Attribute definitions:

SELF\placement.location: A reference point on the axis.

axis: The direction of the local Z axis.

z: The normalised direction of the local Z axis.

SELF\geometric_representation_item.dim: The space dimensionality of the **axis1_placement**, which is determined from its location, and is always equal to 3.

Formal propositions:

WR1: The coordinate space dimensionality shall be 3.

4.4.14 axis2_placement_2d

The location and orientation in two-dimensional space of two mutually perpendicular axes. An **axis2_placement_2d** is defined in terms of a point, (inherited from the placement supertype), and an axis. It can be used to locate and orientate an object in two-dimensional space and to

define a placement coordinate system. The entity includes a point which forms the origin of the placement coordinate system. A direction vector is required to complete the definition of the placement coordinate system. The **ref_direction** defines the placement X axis direction; the placement Y axis direction is derived from this.

EXPRESS specification:

```

*)
ENTITY axis2_placement_2d
  SUBTYPE OF (placement);
  ref_direction : OPTIONAL direction;
  DERIVE
    p          : LIST [2:2] OF direction := build_2axes(ref_direction);
  WHERE
    WR1: SELF\geometric_representation_item.dim = 2;
  END_ENTITY;
  (*)

```

Attribute definitions:

SELF\placement.location: The spatial position of the reference point which defines the origin of the associated placement coordinate system.

ref.direction: The direction used to determine the direction of the local X axis. If **ref.direction** is omitted, this direction is taken from the geometric coordinate system.

p: The axis set for the placement coordinate system.

p[1]: The normalised direction of the placement X axis. This is (1.0,0.0) if **ref.direction** is omitted.

p[2]: The normalised direction of the placement Y axis. This is a derived attribute and is orthogonal to p[1].

Formal propositions:

WR1: The space dimensionality of the **axis2_placement_2d** shall be 2.

4.4.15 axis2_placement_3d

The location and orientation in three-dimensional space of two mutually perpendicular axes. An **axis2_placement_3d** is defined in terms of a point, (inherited from the placement supertype), and two (ideally orthogonal) axes. It can be used to locate and orientate a non axis-symmetric object in space and to define a placement coordinate system. The entity includes a point which forms the origin of the placement coordinate system. Two direction vectors are required to complete the definition of the placement coordinate system. The **axis** is the placement Z axis direction and the **ref_direction** is an approximation to the placement X axis direction.

NOTE - Let **z** be the placement Z axis direction and **a** be the approximate placement X axis direction. There are two methods, mathematically identical but numerically different for calculating the placement X and Y axis directions.

ISO 10303-42:1994(E)

a) The vector $\mathbf{a}$ is projected onto the plane defined by the origin point P and the vector $\mathbf{z}$ to give the placement X axis direction as $\mathbf{x} = \langle \mathbf{a} - (\mathbf{a} \cdot \mathbf{z})\mathbf{z} \rangle$. The placement Y axis direction is then given by $\mathbf{y} = \langle \mathbf{z} \times \mathbf{x} \rangle$.

b) The placement Y axis direction is calculated as $\mathbf{y} = \langle \mathbf{z} \times \mathbf{a} \rangle$ and then the placement X axis direction is given by $\mathbf{x} = \langle \mathbf{y} \times \mathbf{z} \rangle$.

The first method is likely to be the more numerically stable of the two, and is used here.

A placement coordinate system referenced by the parametric equations is derived from the `axis2_placement_3d` data for conic curves and elementary surfaces.

EXPRESS specification:

```
*)
ENTITY axis2_placement_3d
  SUBTYPE OF (placement);
  axis          : OPTIONAL direction;
  ref_direction : OPTIONAL direction;
DERIVE
  p              : LIST [3:3] OF direction := build_axes(axis,ref_direction);
WHERE
  WR1: SELF\placement.location.dim = 3;
  WR2: (NOT (EXISTS (axis))) OR (axis.dim = 3);
  WR3: (NOT (EXISTS (ref_direction))) OR (ref_direction.dim = 3);
  WR4: (NOT (EXISTS (axis))) OR (NOT (EXISTS (ref_direction))) OR
    (cross_product(axis,ref_direction).magnitude > 0.0);
END_ENTITY;
(*
```

Attribute definitions:

SELF\placement.location: The spatial position of the reference point and origin of the associated placement coordinate system.

axis: The exact direction of the local Z axis.

ref_direction: The direction used to determine the direction of the local X axis. If necessary an adjustment is made to maintain orthogonality to the **axis** direction. If **axis** and/or **ref_direction** is omitted, these directions are taken from the geometric coordinate system.

p: The axes for the placement coordinate system. The directions of these axes are derived from the attributes, with appropriate default values if required.

p[1]: The normalised direction of the local X axis.

p[2]: The normalised direction of the local Y axis

p[3]: The normalised direction of the local Z axis.

NOTE – See figure 1 for interpretation of attributes.

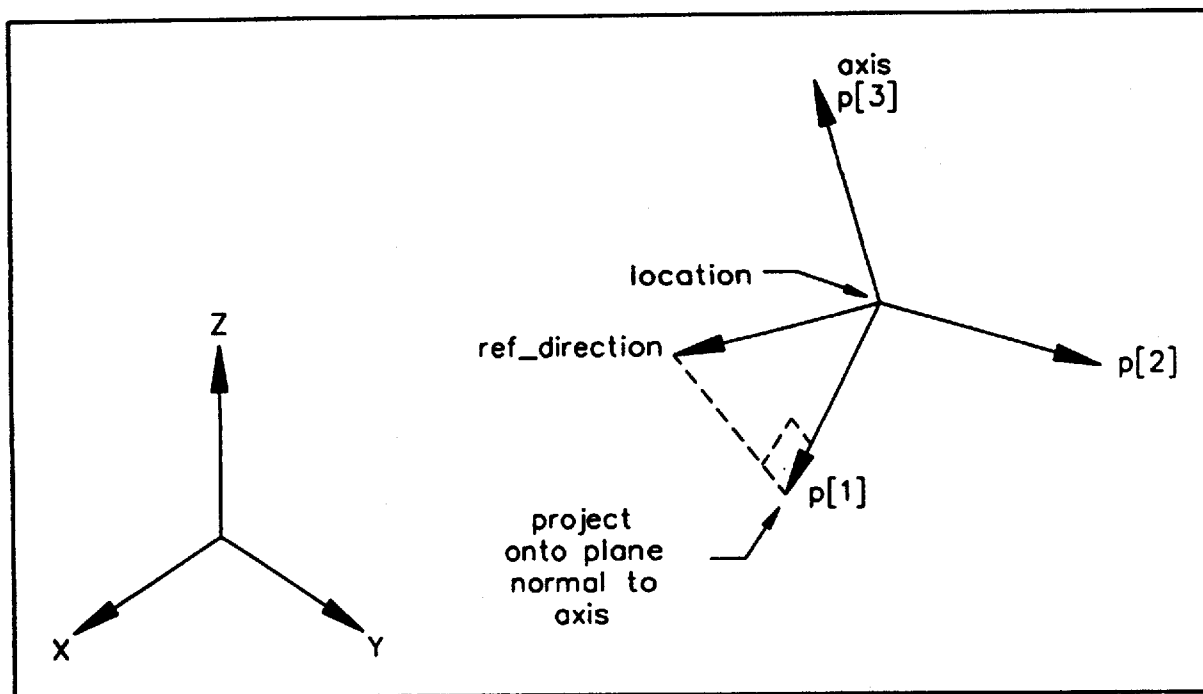


Figure 1 – Axis2 placement 3D

Formal propositions:

WR1: The space dimensionality of the **SELF\placement.location** shall be 3.

WR2: The space dimensionality of **axis** shall be 3.

WR3: The space dimensionality of **ref.direction** shall be 3.

WR4: The **axis** and the **ref.direction** shall not be parallel or anti-parallel. (This is required by the **build_axes** function.)

4.4.16 cartesian_transformation_operator

A **cartesian_transformation_operator** defines a geometric transformation composed of translation, rotation, mirroring and uniform scaling.

The list of normalised vectors **u** defines the columns of an orthogonal matrix **T**. These vectors are computed, by the **base_axis** function, from the direction attributes **axis1**, **axis2** and, in **cartesian_transformation_operator_3d**, **axis3**. If $|T| = -1$, the transformation includes mirroring. The local origin point **A**, the scale value **S** and the matrix **T** together define a transformation.

The transformation for a point with position vector **P** is defined by

$$P \rightarrow A + STP$$

The transformation for a direction **d** is defined by

$$d \rightarrow Td$$

The transformation for a vector with orientation **d** and magnitude **k** is defined by

$$d \rightarrow Td$$

and

$$k \rightarrow Sk$$

For those entities whose attributes include an **axis2_placement**, the transformation is applied, after the derivation, to the derived attributes **p** defining the placement coordinate directions. For a transformed **surface**, the direction of the surface normal at any point is obtained by transforming the normal, at the corresponding point, to the original **surface**. For geometric entities with attributes (such as the radius of a circle) which have the dimensionality of length, the values will be multiplied by **S**.

For curves on surface the **p_curve.reference_to_curve** will be unaffected by any transformation.

The **cartesian_transformation_operator** shall only be applied to geometry defined in a consistent system of units with the same units on each axis. With all attributes omitted, the transformation defaults to the identity transformation. The **cartesian_transformation_operator** shall only be instantiated as one of its subtypes.

NOTE – See figures 2(a-c) for demonstration of effect of transformation.

EXPRESS specification:

```

*)
ENTITY cartesian_transformation_operator
  SUPERTYPE OF (ONEOF(cartesian_transformation_operator_2d,
                       cartesian_transformation_operator_3d))
  SUBTYPE OF (geometric_representation_item,
              functionally_defined_transformation);
  axis1      : OPTIONAL direction;
  axis2      : OPTIONAL direction;
  local_origin : cartesian_point;
  scale      : OPTIONAL REAL;
DERIVE
  scl      : REAL := NVL(scale, 1.0);
WHERE
  WR1: scl > 0.0;
END_ENTITY;
(

```

Attribute definitions:

axis1: The direction used to determine **u[1]**, the derived X axis direction.

axis2: The direction used to determine **u[2]**, the derived Y axis direction.