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

Part 42:

**Integrated generic resource: Geometric
and topological representation**

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

*Partie 42: Ressource générique intégrée: Représentation géométrique
et topologique*



Reference number
ISO 10303-42:2003(E)

AVA Documents
Document Sharing Hub
© ISO 2003

PDF disclaimer

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

Adobe is a trademark of Adobe Systems Incorporated.

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

© ISO 2003

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 either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

1	Scope	1
1.1	Geometry	1
1.2	Topology	2
1.3	Geometric Shape Models	2
2	Normative references	3
3	Terms, definitions, symbols and abbreviations	3
3.1	Terms defined in ISO 10303-1	3
3.2	Other terms and definitions	4
3.3	Symbols	9
3.4	Abbreviations	12
4	Geometry	13
4.1	Introduction	14
4.2	Fundamental concepts and assumptions	14
4.2.1	Space dimensionality	14
4.2.2	Geometric relationships	15
4.2.3	Parametrisation of analytic curves and surfaces	15
4.2.4	Curves	15
4.2.5	Surfaces	15
4.2.6	Preferred form	16
4.3	Geometry constant and type definitions	16
4.3.1	dummy_gri	16
4.3.2	dimension_count	17
4.3.3	b_spline_curve_form	17
4.3.4	b_spline_surface_form	18
4.3.5	extent_enumeration	19
4.3.6	knot_type	19
4.3.7	preferred_surface_curve_representation	20
4.3.8	transition_code	21
4.3.9	trimming_preference	21
4.3.10	axis2_placement	22
4.3.11	curve_on_surface	22
4.3.12	pcurve_or_surface	23
4.3.13	surface_boundary	23
4.3.14	trimming_select	23
4.3.15	vector_or_direction	24
4.4	Geometry entity definitions	24
4.4.1	geometric_representation_context	24
4.4.2	geometric_representation_item	25
4.4.3	point	27
4.4.4	cartesian_point	27

4.4.5	cylindrical_point	28
4.4.6	spherical_point	29
4.4.7	polar_point	30
4.4.8	point_on_curve	31
4.4.9	point_on_surface	32
4.4.10	point_in_volume	32
4.4.11	point_replica	33
4.4.12	degenerate_pcurve	34
4.4.13	evaluated_degenerate_pcurve	35
4.4.14	direction	35
4.4.15	vector	36
4.4.16	placement	37
4.4.17	axis1_placement	37
4.4.18	axis2_placement_2d	38
4.4.19	axis2_placement_3d	39
4.4.20	cartesian_transformation_operator	41
4.4.21	cartesian_transformation_operator_3d	43
4.4.22	cartesian_transformation_operator_2d	46
4.4.23	curve	47
4.4.24	line	47
4.4.25	conic	48
4.4.26	circle	49
4.4.27	ellipse	49
4.4.28	hyperbola	52
4.4.29	parabola	53
4.4.30	clothoid	55
4.4.31	circular_involute	56
4.4.32	bounded_curve	58
4.4.33	polyline	59
4.4.34	b_spline_curve	59
4.4.35	b_spline_curve_with_knots	62
4.4.36	uniform_curve	64
4.4.37	quasi_uniform_curve	65
4.4.38	bezier_curve	65
4.4.39	rational_b_spline_curve	67
4.4.40	trimmed_curve	68
4.4.41	composite_curve	71
4.4.42	composite_curve_segment	73
4.4.43	reparametrised_composite_curve_segment	74
4.4.44	pcurve	75
4.4.45	bounded_pcurve	76
4.4.46	surface_curve	76
4.4.47	intersection_curve	78
4.4.48	seam_curve	79
4.4.49	bounded_surface_curve	79
4.4.50	composite_curve_on_surface	80

4.4.51	offset_curve_2d	81
4.4.52	offset_curve_3d	83
4.4.53	curve_replica	84
4.4.54	surface	85
4.4.55	elementary_surface	85
4.4.56	plane	86
4.4.57	cylindrical_surface	87
4.4.58	conical_surface	88
4.4.59	spherical_surface	90
4.4.60	toroidal_surface	91
4.4.61	degenerate_toroidal_surface	92
4.4.62	dupin_cyclide_surface	94
4.4.63	swept_surface	98
4.4.64	surface_of_linear_extrusion	99
4.4.65	surface_of_revolution	100
4.4.66	surface_curve_swept_surface	101
4.4.67	fixed_reference_swept_surface	103
4.4.68	bounded_surface	105
4.4.69	b_spline_surface	105
4.4.70	b_spline_surface_with_knots	108
4.4.71	uniform_surface	110
4.4.72	quasi_uniform_surface	110
4.4.73	bezier_surface	111
4.4.74	rational_b_spline_surface	112
4.4.75	rectangular_trimmed_surface	113
4.4.76	curve_bounded_surface	115
4.4.77	boundary_curve	117
4.4.78	outer_boundary_curve	117
4.4.79	rectangular_composite_surface	117
4.4.80	surface_patch	119
4.4.81	offset_surface	120
4.4.82	oriented_surface	122
4.4.83	surface_replica	122
4.4.84	volume	123
4.4.85	block_volume	124
4.4.86	wedge_volume	125
4.4.87	pyramid_volume	127
4.4.88	tetrahedron_volume	128
4.4.89	hexahedron_volume	129
4.4.90	spherical_volume	130
4.4.91	cylindrical_volume	131
4.4.92	eccentric_conical_volume	132
4.4.93	toroidal_volume	134
4.4.94	ellipsoid_volume	135
4.4.95	b_spline_volume	135
4.4.96	b_spline_volume_with_knots	138

4.4.97	bezier_volume	139
4.4.98	uniform_volume	140
4.4.99	quasi_uniform_volume	141
4.4.100	rational_b_spline_volume	142
4.5	Geometry schema rule definition: compatible_dimension	143
4.6	Geometry function definitions	144
4.6.1	dimension_of	145
4.6.2	acyclic_curve_replica	146
4.6.3	acyclic_point_replica	147
4.6.4	acyclic_surface_replica	147
4.6.5	associated_surface	148
4.6.6	base_axis	149
4.6.7	build_2axes	150
4.6.8	build_axes	151
4.6.9	orthogonal_complement	152
4.6.10	first_proj_axis	152
4.6.11	second_proj_axis	154
4.6.12	cross_product	155
4.6.13	dot_product	156
4.6.14	normalise	157
4.6.15	scalar_times_vector	158
4.6.16	vector_sum	159
4.6.17	vector_difference	161
4.6.18	default_b_spline_knot_mult	162
4.6.19	default_b_spline_knots	163
4.6.20	default_b_spline_curve_weights	164
4.6.21	default_b_spline_surface_weights	165
4.6.22	constraints_param_b_spline	165
4.6.23	curve_weights_positive	167
4.6.24	constraints_composite_curve_on_surface	168
4.6.25	get_basis_surface	168
4.6.26	surface_weights_positive	169
4.6.27	volume_weights_positive	170
4.6.28	constraints_rectangular_composite_surface	171
4.6.29	list_to_array	172
4.6.30	make_array_of_array	173
4.6.31	make_array_of_array_of_array	174
4.6.32	above_plane	175
4.6.33	same_side	176
5	Topology	178
5.1	Introduction	178
5.2	Fundamental concepts and assumptions	178
5.2.1	Geometric associations	179
5.2.2	Associations with parameter space geometry	180
5.2.3	Graphs, cycles, and traversals	182

5.3	Topology constant and type definitions	183
5.3.1	dummy_tri	183
5.3.2	shell	183
5.3.3	reversible_topology_item	184
5.3.4	list_of_reversible_topology_item	184
5.3.5	set_of_reversible_topology_item	185
5.3.6	reversible_topology	185
5.4	Topology entity definitions	185
5.4.1	topological_representation_item	185
5.4.2	vertex	186
5.4.3	vertex_point	187
5.4.4	edge	187
5.4.5	edge_curve	189
5.4.6	oriented_edge	190
5.4.7	seam_edge	191
5.4.8	subedge	192
5.4.9	path	192
5.4.10	oriented_path	193
5.4.11	open_path	194
5.4.12	loop	195
5.4.13	vertex_loop	196
5.4.14	edge_loop	197
5.4.15	poly_loop	198
5.4.16	face_bound	199
5.4.17	face_outer_bound	199
5.4.18	face	200
5.4.19	face_surface	202
5.4.20	oriented_face	203
5.4.21	subface	204
5.4.22	connected_face_set	205
5.4.23	vertex_shell	206
5.4.24	wire_shell	206
5.4.25	open_shell	208
5.4.26	oriented_open_shell	210
5.4.27	closed_shell	211
5.4.28	oriented_closed_shell	213
5.4.29	connected_face_sub_set	214
5.4.30	connected_edge_set	215
5.5	Topology function definitions	215
5.5.1	conditional_reverse	215
5.5.2	topology_reversed	216
5.5.3	edge_reversed	217
5.5.4	path_reversed	218
5.5.5	face_bound_reversed	218
5.5.6	face_reversed	219
5.5.7	shell_reversed	220

5.5.8	closed_shell_reversed	220
5.5.9	open_shell_reversed	221
5.5.10	set_of_topology_reversed	222
5.5.11	list_of_topology_reversed	223
5.5.12	boolean_choose	223
5.5.13	path_head_to_tail	224
5.5.14	list_face_loops	225
5.5.15	list_loop_edges	225
5.5.16	list_shell_edges	226
5.5.17	list_shell_faces	226
5.5.18	list_shell_loops	227
5.5.19	mixed_loop_type_set	228
5.5.20	list_to_set	229
5.5.21	edge_curve_pcurves	229
5.5.22	vertex_point_pcurves	231
6	Geometric models	232
6.1	Introduction	232
6.2	Fundamental concepts and assumptions	233
6.3	Geometric model type definitions	233
6.3.1	boolean_operand	233
6.3.2	boolean_operator	234
6.3.3	csg_primitive	234
6.3.4	csg_select	235
6.3.5	geometric_set_select	235
6.3.6	surface_model	236
6.3.7	wireframe_model	236
6.4	Geometric model entity definitions	236
6.4.1	solid_model	236
6.4.2	manifold_solid_brep	237
6.4.3	brep_with_voids	240
6.4.4	faceted_brep	240
6.4.5	brep_2d	241
6.4.6	csg_solid	242
6.4.7	boolean_result	243
6.4.8	block	244
6.4.9	right_angular_wedge	245
6.4.10	rectangular_pyramid	246
6.4.11	faceted_primitive	247
6.4.12	tetrahedron	248
6.4.13	convex_hexahedron	249
6.4.14	sphere	250
6.4.15	right_circular_cone	251
6.4.16	right_circular_cylinder	252
6.4.17	eccentric_cone	253
6.4.18	torus	254

6.4.19	ellipsoid	255
6.4.20	cyclide_segment_solid	256
6.4.21	half_space_solid	257
6.4.22	boxed_half_space	259
6.4.23	box_domain	259
6.4.24	primitive_2d	260
6.4.25	circular_area	261
6.4.26	elliptic_area	261
6.4.27	rectangular_area	262
6.4.28	polygonal_area	262
6.4.29	half_space_2d	263
6.4.30	rectangled_half_space	264
6.4.31	rectangle_domain	264
6.4.32	swept_face_solid	265
6.4.33	extruded_face_solid	266
6.4.34	revolved_face_solid	266
6.4.35	surface_curve_swept_face_solid	268
6.4.36	swept_area_solid	270
6.4.37	extruded_area_solid	270
6.4.38	revolved_area_solid	271
6.4.39	surface_curve_swept_area_solid	272
6.4.40	swept_disk_solid	273
6.4.41	trimmed_volume	275
6.4.42	solid_replica	276
6.4.43	shell_based_surface_model	277
6.4.44	face_based_surface_model	277
6.4.45	shell_based_wireframe_model	278
6.4.46	edge_based_wireframe_model	279
6.4.47	geometric_set	280
6.4.48	geometric_curve_set	281
6.4.49	sectioned_spine	281
6.4.50	geometric_set_replica	282
6.5	Geometric model function definitions	283
6.5.1	acyclic_solid_replica	283
6.5.2	acyclic_set_replica	284
6.5.3	constraints_geometry_shell_based_surface_model	285
6.5.4	constraints_geometry_shell_based_wireframe_model	285
6.5.5	build_transformed_set	286
6.5.6	msb_shells	287
Annex A (normative)	Short names of entities	289
Annex B (normative)	Information object registration	297
B.1	Document identification	297
B.2	Schema identification	297
Annex C (informative)	Computer-interpretable listings	298

Annex D (informative) EXPRESS-G diagrams	299
Bibliography	320
Index	321

Figures

Figure 1	Spherical_point attributes	30
Figure 2	Axis2_placement_3d	41
Figure 3	(a) Cartesian_transformation_operator_3d	43
Figure 3	(b) Cartesian_transformation_operator_3d	44
Figure 3	(c) Cartesian_transformation_operator_3d	45
Figure 4	Circle	50
Figure 5	Ellipse	51
Figure 6	Hyperbola	53
Figure 7	Parabola	54
Figure 8	Clothoid curve	57
Figure 9	Circular involute curve	58
Figure 10	B-spline curve	61
Figure 11	Composite_curve	72
Figure 12	Conical_surface	89
Figure 13	Cross section of degenerate_toroidal_surface	93
Figure 14	Cross-sections of a Dupin cyclide with $C = 0$	96
Figure 15	A Dupin ring cyclide	97
Figure 16	A Dupin horned cyclide	98
Figure 17	A Dupin spindle cyclide	99
Figure 18	Fixed_reference_swept_surface	104
Figure 19	Curve bounded surface	116
Figure 20	Wedge_volume and its attributes	125
Figure 21	Edge curve	188
Figure 22	Right angular wedge and its attributes	245
Figure 23	Convex_hexahedron	251
Figure 24	Cyclide_segment_solid	257
Figure 25	Cross section of cyclide_segment_solid	258
Figure 26	Revolved face solid	267
Figure D.1	Geometry_schema EXPRESS-G diagram 1 of 13	300
Figure D.2	Geometry_schema EXPRESS-G diagram 2 of 13	301
Figure D.3	Geometry_schema EXPRESS-G diagram 3 of 13	302
Figure D.4	Geometry_schema EXPRESS-G diagram 4 of 13	303
Figure D.5	Geometry_schema EXPRESS-G diagram 5 of 13	304
Figure D.6	Geometry_schema EXPRESS-G diagram 6 of 13	305
Figure D.7	Geometry_schema EXPRESS-G diagram 7 of 13	306
Figure D.8	Geometry_schema EXPRESS-G diagram 8 of 13	307
Figure D.9	Geometry_schema EXPRESS-G diagram 9 of 13	308
Figure D.10	Geometry_schema EXPRESS-G diagram 10 of 13	309

Figure D.11	Geometry_schema EXPRESS-G diagram 11 of 13	310
Figure D.12	Geometry_schema EXPRESS-G diagram 12 of 13	311
Figure D.13	Geometry_schema EXPRESS-G diagram 13 of 13	312
Figure D.14	Topology_schema EXPRESS-G diagram 1 of 3	313
Figure D.15	Topology_schema EXPRESS-G diagram 2 of 3	314
Figure D.16	Topology_schema EXPRESS-G diagram 3 of 3	315
Figure D.17	Geometric_model_schema EXPRESS-G diagram 1 of 4	316
Figure D.18	Geometric_model_schema EXPRESS-G diagram 2 of 4	317
Figure D.19	Geometric_model_schema EXPRESS-G diagram 3 of 4	318
Figure D.20	Geometric_model_schema EXPRESS-G diagram 4 of 4	319

Tables

Table 1	Geometry mathematical symbology	9
Table 2	Topology symbol definitions	11
Table A.1	Short names of entities	289

Foreword

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

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

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

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

ISO 10303-42 was prepared by Technical Committee ISO / TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

This third edition cancels and replaces the second edition (ISO 10303-42:2000) of which it constitutes a minor technical revision. The first edition (ISO 10303-42:1994) is provisionally retained to support continued use and maintenance of implementations based on the first edition and to satisfy normative references in other parts of ISO 10303.

The corrections published in ISO 10303-42:2000/Cor.1:2001 are incorporated in this edition.

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

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

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

[<http://www.tc184-sc4.org/titles/STEP_titles.rtf>](http://www.tc184-sc4.org/titles/STEP_titles.rtf)

Should further parts of ISO 10303 be published, they will follow the same numbering pattern.

This part of ISO 10303 is a member of the integrated resources series. The integrated resources specify a single conceptual product data model.

Introduction

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

This part of ISO 10303 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.

This edition incorporates modifications that are upwardly compatible with the previous editions. Modifications to EXPRESS specifications are upwardly compatible if:

- instances encoded according to ISO 10303-21 and that conform to an ISO 10303 application protocol based on the previous edition of this part, also conform to a revision of that application protocol based on this edition;
- interfaces that conform to ISO 10303-22 and to an ISO 10303 application protocol based on the previous edition of this part, also conform to a revision of that application protocol based on this edition;
- the mapping tables of ISO 10303 application protocols based on the previous edition of this part remain valid in a revision of that application protocol based on this edition.

Technical modifications to ISO 10303-42:1994 are categorised as follows:

ISO 10303-42:2003(E)

- changes to the EXPRESS declarations,
- new EXPRESS declarations.

The following EXPRESS declarations were modified in creating edition 2:

geometry schema:

- **axis1_placement;**
- **base_axis;**
- **build_axes;**
- **build_2axes;**
- **cartesian_transformation_operator_3d;**
- **cartesian_transformation_operator_2d;**
- **composite_curve_segment;**
- **constraints_param_b_spline;**
- **cross_product;**
- **curve_bounded_surface;**
- **default_b_spline_curve_weights;**
- **default_b_spline_knot_mult;**
- **default_b_spline_knots;**
- **default_b_spline_surface_weights;**
- **geometric_representation_item;**
- **get_basis_surface;**
- **list_to_array;**
- **make_array_of_array;**
- **orthogonal_complement;**
- **point;**

- **rectangular_composite_surface;**
- **scalar_times_vector;**
- **surface_of_revolution;**
- **surface_patch;**
- **swept_surface;**
- **trimmed_curve;**
- **vector_sum;**
- **vector_difference;**

topology schema:

- **edge;**
- **edge_reversed;**
- **face_bound_reversed;**
- **face_reversed;**
- **face_surface;**
- **mixed_loop_type_set;**
- **path_head_to_tail;**
- **path_reversed;**
- **shell_reversed;**

geometric model schema:

- **boolean_operand;**
- **build_transformed_set;**
- **csg_primitive;**
- **csg_solid;**

ISO 10303-42:2003(E)

- **revolved_area_solid;**
- **revolved_face_solid;**
- **solid_model;**
- **swept_area_solid;**
- **swept_face_solid.**

The following EXPRESS declarations were added in edition 2:

geometry schema:

- **above_plane;**
- **b_spline_volume;**
- **b_spline_volume_with_knots;**
- **bezier_volume;**
- **block_volume;**
- **clothoid;**
- **cylindrical_point;**
- **cylindrical_volume;**
- **dummy_gri;**
- **dupin_cyclide_surface;**
- **eccentric_conical_volume;**
- **ellipsoid_volume;**
- **oriented_surface;**
- **hexahedron_volume;**
- **make_array_of_array_of_array;**
- **point_in_volume;**
- **polar_point;**

- **pyramid_volume;**
- **quasi_uniform_volume;**
- **rational_b_spline_volume;**
- **same_side;**
- **spherical_point;**
- **spherical_volume;**
- **surface_boundary;**
- **surface_curve_swept_surface;**
- **tetrahedron_volume;**
- **toroidal_volume;**
- **volume;**
- **wedge_volume;**

topology schema:

- **closed_shell_reversed;**
- **connected_face_sub_set;**
- **dummy_tri;**
- **open_shell_reversed;**
- **seam_edge;**
- **subedge;**

geometric model schema:

- **brep_2d;**
- **circular_area;**
- **convex_hexahedron;**

ISO 10303-42:2003(E)

- **cyclide_segment_solid;**
- **eccentric_cone;**
- **ellipsoid;**
- **elliptic_area;**
- **faceted_primitive;**
- **half_space_2d;**
- **polygonal_area;**
- **primitive_2d;**
- **rectangular_area;**
- **rectangular_pyramid;**
- **sectioned_spine;**
- **surface_curve_swept_area_solid;**
- **surface_curve_swept_face_solid;**
- **tetrahedron;**
- **trimmed_volume.**

Technical corrigendum 1 modified the following EXPRESS declarations:

- **surface_of_revolution** (geometry schema)
- **first_proj_axis** (geometry schema)
- **list_to_array** (FUNCTION geometry schema)
- **make_array_of_array** (FUNCTION geometry schema)
- **make_array_of_array_of_array** (FUNCTION geometry schema)
- **revolved_face_solid** (geometric_model schema)
- **revolved_area_solid** (geometric_model schema)
- **box_domain** (geometric_model schema)

- **rectangle_domain** (geometric_model schema)
- **build_transformed_set** (FUNCTION geometric_model schema)

This revision (edition 3) corrects the definition of **path_head_to_tail** function and adds the following new EXPRESS declarations:

- **circular_involute** (geometry schema)
- **swept_disk_solid** (geometric_model schema)

Several components of this part of ISO 10303 are available in electronic form. This access is provided through the specification of Universal Resource Locators (URL's) that identify the location of these files on the internet. If there is difficulty in accessing these files, contact the ISO Central Secretariat or the ISO SC4 Secretariat directly at: sc4sec@tc184-sc4.org.

Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: 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 finite volumes with internal parametrisation;
- 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, surfaces and volumes;
- definition of point, curve and surface replicas;
- definition of offset curves and surfaces;
- definition of intersection curves.

ISO 10303-42:2003(E)

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

- all other forms of procedurally defined curves and surfaces;
- 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;
- CSG models in two-dimensional space;
- 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 referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8824-1:1998, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation*

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

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

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

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

3 Terms, definitions, symbols and abbreviations

3.1 Terms defined in ISO 10303-1

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

- integrated resource.

3.2 Other terms and definitions

For the purposes of this part of ISO 10303, the following terms and definitions apply. A number of informal definitions are also given here which will later be used to describe and constrain the topological entities. They are not intended to be mathematically rigorous. The definitions are given in alphabetical, not logical order.

3.2.1

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.

3.2.2

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.

3.2.3

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.

3.2.4

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 .

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

NOTE 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.

3.2.5

boundary representation solid model (B-rep)

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.

3.2.6

closed curve

a curve such that both end points are the same.

3.2.7

closed surface

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

3.2.8**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.

3.2.9**connected**

equivalent to **arcwise connected** (see 3.2.1).

3.2.10**connected component**

a maximal connected subset of a domain.

3.2.11**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.

3.2.12**coordinate space**

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

3.2.13**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.

3.2.14**cycle**

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

3.2.15 **d -manifold with boundary**

a domain which is the union of its d -dimensional interior and its boundary.

3.2.16**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.

3.2.17**domain**

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

3.2.18

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 5.4.25 and 5.4.27.

3.2.19

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.

3.2.20

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.

3.2.21

genus of a graph

the integer-valued invariant defined algorithmically by the graph traversal algorithm described in the note in 5.2.3.

3.2.22

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.

3.2.23

geometrically founded

a property of **geometric_representation_items** (see 4.4.2) 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.

3.2.24

geometrically related

the relationship between two **geometric_representation_items** (see 4.4.2) in the same context by which the concepts of distance and direction between them are defined.

3.2.25

geometric coordinate system

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

3.2.26**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.

3.2.27**handle**

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

3.2.28**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.

3.2.29**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.

3.2.30**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 .

3.2.31**list**

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

3.2.32**model space**

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

3.2.33**open curve**

a curve which has two distinct end points.

3.2.34**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.

ISO 10303-42:2003(E)

3.2.35

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.

3.2.36

overlap

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

3.2.37

parameter range

the range of valid parameter values for a curve, surface, or volume.

3.2.38

parameter space

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

3.2.39

parametric volume

a bounded region of three dimensional model space with an associated parametric coordinate system such that every interior point is associated with a list (u, v, w) of parameter values.

3.2.40

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.

3.2.41

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.

3.2.42

self-loop

an edge that has the same vertex at both ends.

3.2.43

set

an unordered collection in which no two members are equal.

3.2.44**space dimensionality**

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

3.2.45**surface**

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

3.2.46**topological sense**

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

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

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

3.3 Symbols

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

3.3.1 Geometry and mathematical symbology

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

Table 1 – Geometry mathematical symbology

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

3.3.2 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 1 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 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_l or S^o and S_l^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:

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

Thus

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

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

3.4 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
    (definitional_representation,
     founded_item,
     functionally_defined_transformation,
     item_in_context,
     representation,
     representation_item,
     representation_context,
     using_representations);
  REFERENCE FROM measure_schema
    (global_unit_assigned_context,
     length_measure,
     parameter_value,
     plane_angle_measure,
     plane_angle_unit,
     positive_length_measure,
     positive_plane_angle_measure);
  REFERENCE FROM topology_schema
    (edge_curve,
     face_surface,
     poly_loop,
     vertex_point);
  REFERENCE FROM geometric_model_schema
    (block,
     boolean_result,
     cyclide_segment_solid,
     eccentric_cone,
     edge_based_wireframe_model,
     ellipsoid,
     face_based_surface_model,
     faceted_primitive,
     geometric_set,
     half_space_solid,
     half_space_2d,
     primitive_2d,
     rectangular_pyramid,
     right_angular_wedge,
     right_circular_cone,
     right_circular_cylinder,
     shell_based_surface_model,
     shell_based_wireframe_model,

```

```
solid_model,  
sphere,  
torus);  
(*
```

NOTE 1 The schemas referenced above can be found in the following parts of ISO 10303:

representation_schema	ISO 10303-43
measure_schema	ISO 10303-41
topology_schema	clause 5 of this part of ISO 10303
geometric_model_schema	clause 6 of this part of ISO 10303

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

NOTE 3 See annex D, Figures D.1 to D.13, 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. The **geometric_representation_context** enables a distinction to be made between those items which are in the same context, and thus 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 a 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. The geometric direction of a curve is the direction of increasing 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 parametri-

sation. 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. The **rectangular_composite_surface** entity provides the basic 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 constant and type definitions

4.3.1 dummy_gri

The constant **dummy_gri** is a partial entity definition to be used when types of **geometric_representation_item** are constructed. It provides the correct supertypes and the **name** attribute as an empty string.

EXPRESS specification:

```
* )
  CONSTANT
    dummy_gri : geometric_representation_item := representation_item('') ||
                                              geometric_representation_item();
  END_CONSTANT;
( *
```

4.3.2 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.

4.3.3 b_spline_curve_form

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

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.

4.3.4 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.

4.3.5 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.

4.3.6 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 (4.4.35).

EXPRESS specification:

```
* )
```

ISO 10303-42:2003(E)

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

4.3.8 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.

4.3.9 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: Trimming by cartesian point is preferred.

parameter: Trimming by parameter value is preferred.

unspecified: No trimming preference is communicated.

4.3.10 axis2_placement

This select type represents the placing of mutually perpendicular axes in two-dimensional or in three-dimensional Cartesian space.

NOTE This select type enables entities requiring axis placement information to reference the axes without specifying the space dimensionality.

EXPRESS specification:

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

4.3.11 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;
( *

```

4.3.12 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;
( *

```

4.3.13 surface_boundary

This type is used to select the type of bounding curve to be used in the definition of a **curve_bounded_surface**. It provides for the boundary to be either a **boundary_curve** or a **degenerate_pcurve**.

EXPRESS specification:

```

*)
TYPE surface_boundary = SELECT
    (boundary_curve,
     degenerate_pcurve);
END_TYPE;
( *

```

4.3.14 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;  
(*
```

4.3.15 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 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** (see 4.4.2). 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 3.2 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 number of dimensions 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**.

EXAMPLE 1 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**.

EXAMPLE 2 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**.

NOTE 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**.

NOTE 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_repre-**

sensation_context. When in the context of geometry, this relationship causes the **geometric_representation_item** to be geometrically founded.

NOTE 4 As subtypes of **representation_item** there is an implicit and/or relationship between **geometric_representation_item** and **topological_representation_item**. The only complex instances intended to be created are **edge_curve**, **face_surface**, and **vertex_point**.

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, primitive_2d,
    right_angular_wedge, ellipsoid, faceted_primitive,
    rectangular_pyramid, cyclide_segment_solid, volume,
    half_space_solid, half_space_2d,
    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**.

Formal propositions:

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

NOTE 5 The **dim** attribute is derived from the **coordinate_space_dimension** of a **geometric_representation_context** in which the **geometric_representation_item** is geometrically founded.

NOTE 6 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.

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_in_volume, 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 1 For the purposes of defining geometry in this part of ISO 10303 only two or three-dimensional points are used.

NOTE 2 Depending upon the **geometric_representation_context** in which the point is used the names of the coordinates may be (x,y,z), or (u,v), or any other chosen values.

EXPRESS specification:

```
*)
ENTITY cartesian_point
  SUPERTYPE OF (ONEOF(cylindrical_point, polar_point, spherical_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 cylindrical_point

A **cylindrical_point** is a type of **cartesian_point** which uses a cylindrical polar coordinate system, centred at the origin of the corresponding Cartesian coordinate system, to define its location.

EXPRESS specification:

```
*)
ENTITY cylindrical_point
  SUBTYPE OF (cartesian_point);
    r      : length_measure;
    theta  : plane_angle_measure;
    z      : length_measure;
  DERIVE
    SELF\cartesian_point.coordinates : LIST [1:3] OF length_measure :=
      [r*cos(theta), r*sin(theta), z];
  WHERE
    WR1: r >= 0.0;
END_ENTITY;
(*
```

Attribute definitions:

r: The distance from the point to the z axis.

theta: The angle between the plane containing the point and the z axis and the xz plane.

z: The distance from the xy plane to the point.

Formal propositions:

WR1: The radius r shall be greater than, or equal to zero.

Informal propositions:

IP1: The value of **theta** shall lie in the range $0 \leq \text{theta} < 360$ degrees.

4.4.6 spherical_point

A **spherical_point** is a type of **cartesian_point** which uses a spherical polar coordinate system, centred at the origin of the corresponding Cartesian coordinate system, to define its location.

EXPRESS specification:

```

*)
ENTITY spherical_point
  SUBTYPE OF (cartesian_point);
    r      : length_measure;
    theta  : plane_angle_measure;
    phi    : plane_angle_measure;
  DERIVE
    SELF\cartesian_point.coordinates : LIST [1:3] OF length_measure :=
      [r*sin(theta)*cos(phi), r*sin(theta)*sin(phi), r*cos(theta)];
  WHERE
    WR1: r >= 0.0;
END_ENTITY;
( *

```

Attribute definitions:

r: The distance from the point to the origin.

theta: The angle θ between the z axis and the line joining the origin to the point.

phi: The angle ϕ , measured from the x axis to the projection onto the xy plane of the line from the origin to the point.

NOTE See Figure 1 for an illustration of the attributes.

Formal propositions:

WR1: The radius r shall be greater than, or equal to zero.

Informal propositions:

IP1: The value of **theta** shall lie in the range $0 \leq \text{theta} \leq 180$ degrees.

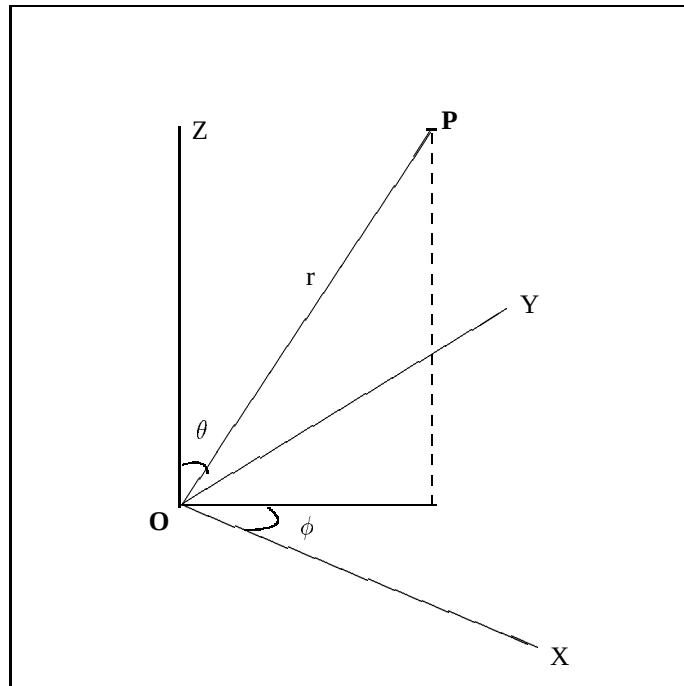


Figure 1 – Spherical_point attributes

IP2: The value of **phi** shall lie in the range $0 \leq \mathbf{phi} < 360$ degrees.

4.4.7 polar_point

A **polar_point** is a type of **cartesian_point** which uses a two dimensional polar coordinate system, centred at the origin of the corresponding Cartesian coordinate system, to define its location.

EXPRESS specification:

```

*)
ENTITY polar_point
  SUBTYPE OF (cartesian_point);
    r      : length_measure;
    theta  : plane_angle_measure;
  DERIVE
    SELF\cartesian_point.coordinates : LIST [1:3] OF length_measure :=
      [r*cos(theta), r*sin(theta)];
  WHERE
    WR1: r >= 0.0;
END_ENTITY;
(

```