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

Part 41:

**Integrated generic resource:
Fundamentals of product description and
support**

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

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



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.

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-41:2000

© ISO 2000

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.ch
Web www.iso.ch

Printed in Switzerland

Contents

	Page
1 Scope	1
1.1 Generic product description resources	1
1.2 Generic management resources	2
1.3 Support resources	2
2 Normative references	2
3 Terms, definitions, and abbreviations	3
3.1 Terms defined in ISO 10303-1	3
3.2 Terms defined in ISO 8601	4
3.3 Other terms and definitions	4
3.4 Abbreviations	5
4 Application context	5
4.1 Introduction	6
4.2 Fundamental concepts and assumptions	6
4.3 Application context entity definitions	6
4.3.1 application_context	6
4.3.2 application_context_element	7
4.3.3 application_context_relationship	8
4.3.4 application_protocol_definition	9
4.3.5 library_context	9
4.3.6 product_concept_context	10
4.3.7 product_context	10
4.3.8 product_definition_context	11
5 Product definition	12
5.1 Introduction	13
5.2 Fundamental concepts and assumptions	13
5.3 Product definition type definition: source	14
5.4 Product definition entity definitions	14
5.4.1 product	14
5.4.2 product_category	15
5.4.3 product_category_relationship	16
5.4.4 product_definition	18
5.4.5 product_definition_context_association	19
5.4.6 product_definition_context_role	20
5.4.7 product_definition_effectivity	20
5.4.8 product_definition_formation	21
5.4.9 product_definition_formation_relationship	22
5.4.10 product_definition_formation_with_specified_source	23
5.4.11 product_definition_relationship	23
5.4.12 product_definition_substitute	24
5.4.13 product_definition_with_associated_documents	26
5.4.14 product_related_product_category	26
5.4.15 product_relationship	26
5.5 Product definition function definitions	27
5.5.1 acyclic_product_category_relationship	27
5.5.2 acyclic_product_definition_formation_relationship	28
5.5.3 acyclic_product_definition_relationship	29
5.5.4 acyclic_product_relationship	30

5.5.5	get_product_definitions	31
6	Product property definition	32
6.1	Introduction	33
6.2	Fundamental concepts and assumptions	33
6.3	Product property definition type definitions	33
6.3.1	characterized_definition	33
6.3.2	characterized_product_definition	34
6.3.3	derived_property_select	34
6.3.4	shape_definition	34
6.4	Product property definition entity definitions	35
6.4.1	characterized_object	35
6.4.2	characterized_object_relationship	35
6.4.3	general_property	36
6.4.4	general_property_association	37
6.4.5	general_property_relationship	38
6.4.6	product_definition_shape	38
6.4.7	property_definition	39
6.4.8	shape_aspect	40
6.4.9	shape_aspect_relationship	41
6.5	Product property definition function definitions	43
6.5.1	acyclic_characterized_object_relationship	43
6.5.2	acyclic_general_property_relationship	44
6.5.3	acyclic_shape_aspect_relationship	45
6.5.4	get_shape_aspects	46
7	Product property representation	47
7.1	Introduction	48
7.2	Fundamental concepts and assumptions	48
7.3	Product property representation type definition: represented_definition	48
7.4	Product property representation entity definitions	49
7.4.1	context_dependent_shape_representation	49
7.4.2	item_identified_representation_usage	50
7.4.3	property_definition_representation	51
7.4.4	shape_definition_representation	52
7.4.5	shape_representation	52
7.4.6	shape_representation_relationship	53
7.5	Product property representation function definitions	53
7.5.1	relatives_of_product_definition	53
7.5.2	relatives_of_shape_representations	54
7.5.3	get_property_definition_representations	55
8	Management resources	56
8.1	Introduction	57
8.2	Fundamental concepts and assumptions	57
8.3	Management resources type definition: attribute_type	57
8.4	Management resources entity definitions	58
8.4.1	action_assignment	58
8.4.2	action_method_assignment	58
8.4.3	action_method_role	59
8.4.4	action_request_assignment	59
8.4.5	approval_assignment	60
8.4.6	attribute_classification_assignment	61
8.4.7	attribute_value_assignment	61

8.4.8	attribute_value_role	62
8.4.9	certification_assignment	62
8.4.10	classification_assignment	63
8.4.11	classification_role	64
8.4.12	contract_assignment	64
8.4.13	date_and_time_assignment	65
8.4.14	date_assignment	65
8.4.15	document_reference	66
8.4.16	document_usage_constraint_assignment	66
8.4.17	document_usage_role	67
8.4.18	effectivity_assignment	68
8.4.19	effectivity_context_assignment	68
8.4.20	effectivity_context_role	69
8.4.21	event_occurrence_assignment	69
8.4.22	event_occurrence_context_assignment	70
8.4.23	external_identification_assignment	70
8.4.24	external_referent_assignment	70
8.4.25	group_assignment	71
8.4.26	identification_assignment	72
8.4.27	identification_assignment_relationship	72
8.4.28	identification_role	73
8.4.29	library_assignment	74
8.4.30	name_assignment	74
8.4.31	organization_assignment	75
8.4.32	organizational_project_assignment	75
8.4.33	organizational_project_role	76
8.4.34	person_and_organization_assignment	76
8.4.35	person_assignment	76
8.4.36	security_classification_assignment	77
8.4.37	time_assignment	78
8.4.38	time_interval_assignment	78
8.5	Management resources function definition: acyclic_identification_assignment_relationship	78
9	Document	80
9.1	Introduction	80
9.2	Fundamental concepts and assumptions	80
9.3	Document type definition: product_or_formation_or_definition	81
9.4	Document entity definitions	81
9.4.1	document	81
9.4.2	document_product_association	82
9.4.3	document_relationship	83
9.4.4	document_representation_type	83
9.4.5	document_type	84
9.4.6	document_usage_constraint	84
9.4.7	document_with_class	85
9.5	Document function definition: acyclic_document_relationship	85
10	Action	86
10.1	Introduction	87
10.2	Fundamental concepts and assumptions	87
10.3	Action type definition: supported_item	87
10.4	Action entity definitions	88
10.4.1	action	88

10.4.2	action_directive	89
10.4.3	action_method	90
10.4.4	action_method_relationship	90
10.4.5	action_relationship	91
10.4.6	action_request_solution	92
10.4.7	action_request_status	92
10.4.8	action_resource	93
10.4.9	action_resource_relationship	93
10.4.10	action_resource_type	94
10.4.11	action_status	94
10.4.12	directed_action	95
10.4.13	executed_action	95
10.4.14	versioned_action_request	96
10.5	Action function definitions	96
10.5.1	acyclic_action_method_relationship	96
10.5.2	acyclic_action_relationship	97
10.5.3	acyclic_action_resource_relationship	98
11	Certification	99
11.1	Introduction	100
11.2	Fundamental concepts and assumptions	100
11.3	Certification entity definitions	100
11.3.1	certification	100
11.3.2	certification_type	101
12	Approval	101
12.1	Introduction	102
12.2	Fundamental concepts and assumptions	102
12.3	Approval entity definitions	102
12.3.1	approval	102
12.3.2	approval_date_time	103
12.3.3	approval_person_organization	104
12.3.4	approval_relationship	104
12.3.5	approval_role	105
12.3.6	approval_status	105
12.4	Approval function definition: acyclic_approval_relationship	106
13	Contract	107
13.1	Introduction	107
13.2	Fundamental concepts and assumptions	108
13.3	Contract entity definitions	108
13.3.1	contract	108
13.3.2	contract_relationship	108
13.3.3	contract_type	109
13.4	Contract function definition: acyclic_contract_relationship	109
14	Security classification	110
14.1	Introduction	111
14.2	Fundamental concepts and assumptions	111
14.3	Security classification entity definitions	111
14.3.1	security_classification	111
14.3.2	security_classification_level	112
15	Person organization	112

15.1	Introduction	113
15.2	Fundamental concepts and assumptions	113
15.3	Person organization type definition: person_organization_select	113
15.4	Person organization entity definitions	114
15.4.1	address	114
15.4.2	organization	115
15.4.3	organization_relationship	116
15.4.4	organization_role	116
15.4.5	organizational_address	117
15.4.6	organizational_project	118
15.4.7	organizational_project_relationship	118
15.4.8	person	119
15.4.9	person_and_organization	120
15.4.10	person_and_organization_role	121
15.4.11	person_role	122
15.4.12	personal_address	123
15.5	Person organization function definitions	123
15.5.1	acyclic_organization_relationship	123
15.5.2	acyclic_organizational_project_relationship	124
16	Date time	125
16.1	Introduction	126
16.2	Fundamental concepts and assumptions	126
16.3	Date time type definitions	126
16.3.1	ahead_or_behind	126
16.3.2	date_time_or_event_occurrence	126
16.3.3	date_time_select	127
16.3.4	day_in_month_number	127
16.3.5	day_in_week_number	127
16.3.6	day_in_year_number	127
16.3.7	hour_in_day	128
16.3.8	minute_in_hour	128
16.3.9	month_in_year_number	128
16.3.10	second_in_minute	129
16.3.11	week_in_year_number	129
16.3.12	year_number	130
16.4	Date time entity definitions	130
16.4.1	calendar_date	130
16.4.2	coordinated_universal_time_offset	131
16.4.3	date	132
16.4.4	date_and_time	132
16.4.5	date_role	132
16.4.6	date_time_role	133
16.4.7	event_occurrence	134
16.4.8	event_occurrence_context_role	134
16.4.9	event_occurrence_relationship	135
16.4.10	event_occurrence_role	135
16.4.11	local_time	136
16.4.12	ordinal_date	137
16.4.13	relative_event_occurrence	137
16.4.14	time_interval	137
16.4.15	time_interval_relationship	138
16.4.16	time_interval_role	139
16.4.17	time_interval_with_bounds	139

16.4.18	time_role	140
16.4.19	week_of_year_and_day_date	141
16.5	Date time function definitions	141
16.5.1	acyclic_event_occurrence_relationship	141
16.5.2	acyclic_time_interval_relationship	142
16.5.3	leap_year	143
16.5.4	valid_calendar_date	143
16.5.5	valid_time	144
17	Group	145
17.1	Introduction	145
17.2	Fundamental concepts and assumptions	145
17.3	Group entity definitions	145
17.3.1	group	145
17.3.2	group_relationship	146
17.4	Group function definition: acyclic_group_relationship	147
18	Effectivity	148
18.1	Introduction	149
18.2	Fundamental concepts and assumptions	149
18.3	Effectivity entity definitions	149
18.3.1	dated_effectivity	149
18.3.2	effectivity	150
18.3.3	effectivity_relationship	151
18.3.4	lot_effectivity	151
18.3.5	serial_numbered_effectivity	152
18.3.6	time_interval_based_effectivity	152
18.4	Effectivity function definition: acyclic_effectivity_relationship	153
19	External reference	154
19.1	Introduction	154
19.2	Fundamental concepts and assumptions	154
19.3	External reference type definitions	155
19.3.1	message	155
19.3.2	source_item	155
19.4	External reference entity definitions	155
19.4.1	external_source	155
19.4.2	external_source_relationship	156
19.4.3	externally_defined_item	157
19.4.4	externally_defined_item_relationship	157
19.4.5	pre_defined_item	158
19.5	External reference function definitions	158
19.5.1	acyclic_external_source_relationship	158
19.5.2	acyclic_externally_defined_item_relationship	159
20	Support resource	160
20.1	Introduction	161
20.2	Fundamental concepts and assumptions	161
20.3	Support resource type definitions	161
20.3.1	identifier	161
20.3.2	label	161
20.3.3	text	162
20.4	Support resource function definitions	162
20.4.1	bag_to_set	162

20.4.2	type_check_function	162
21	Measure	164
21.1	Introduction	164
21.2	Fundamental concepts and assumptions	164
21.3	Measure type definitions	165
21.3.1	amount_of_substance_measure	165
21.3.2	area_measure	165
21.3.3	celsius_temperature_measure	165
21.3.4	context_dependent_measure	165
21.3.5	count_measure	165
21.3.6	descriptive_measure	166
21.3.7	electric_current_measure	166
21.3.8	length_measure	166
21.3.9	luminous_intensity_measure	166
21.3.10	mass_measure	166
21.3.11	measure_value	167
21.3.12	numeric_measure	167
21.3.13	parameter_value	167
21.3.14	plane_angle_measure	168
21.3.15	positive_length_measure	168
21.3.16	positive_plane_angle_measure	168
21.3.17	positive_ratio_measure	168
21.3.18	ratio_measure	169
21.3.19	si_prefix	169
21.3.20	si_unit_name	170
21.3.21	solid_angle_measure	172
21.3.22	thermodynamic_temperature_measure	172
21.3.23	time_measure	172
21.3.24	unit	172
21.3.25	volume_measure	173
21.4	Measure entity definitions	173
21.4.1	amount_of_substance_measure_with_unit	173
21.4.2	amount_of_substance_unit	173
21.4.3	area_measure_with_unit	174
21.4.4	area_unit	174
21.4.5	celsius_temperature_measure_with_unit	175
21.4.6	context_dependent_unit	175
21.4.7	conversion_based_unit	175
21.4.8	derived_unit	176
21.4.9	derived_unit_element	177
21.4.10	dimensional_exponents	177
21.4.11	electric_current_measure_with_unit	178
21.4.12	electric_current_unit	178
21.4.13	global_unit_assigned_context	179
21.4.14	length_measure_with_unit	179
21.4.15	length_unit	179
21.4.16	luminous_intensity_measure_with_unit	180
21.4.17	luminous_intensity_unit	180
21.4.18	mass_measure_with_unit	181
21.4.19	mass_unit	181
21.4.20	measure_with_unit	182
21.4.21	named_unit	182
21.4.22	plane_angle_measure_with_unit	183

21.4.23	plane_angle_unit	183
21.4.24	ratio_measure_with_unit	184
21.4.25	ratio_unit	184
21.4.26	si_unit	184
21.4.27	solid_angle_measure_with_unit	185
21.4.28	solid_angle_unit	185
21.4.29	thermodynamic_temperature_measure_with_unit	186
21.4.30	thermodynamic_temperature_unit	186
21.4.31	time_measure_with_unit	186
21.4.32	time_unit	187
21.4.33	volume_measure_with_unit	187
21.4.34	volume_unit	188
21.5	Measure function definitions	188
21.5.1	derive_dimensional_exponents	188
21.5.2	dimensions_for_si_unit	189
21.5.3	valid_units	190
22	Basic attribute	192
22.1	Introduction	194
22.2	Fundamental concepts and assumptions	195
22.3	Basic attribute type definitions	195
22.3.1	description_attribute_select	195
22.3.2	id_attribute_select	196
22.3.3	name_attribute_select	196
22.3.4	role_select	196
22.4	Basic attribute entity definitions	197
22.4.1	description_attribute	197
22.4.2	id_attribute	197
22.4.3	name_attribute	198
22.4.4	object_role	198
22.4.5	role_association	199
22.5	Basic attribute function definitions	199
22.5.1	get_description_value	199
22.5.2	get_id_value	200
22.5.3	get_name_value	200
22.5.4	get_role	201
Annex A (normative)	Short names of entities	202
Annex B (normative)	Information object registration	208
B.1	Document identification	208
B.2	Schema identification	208
B.2.1	application_context_schema identification	208
B.2.2	product_definition_schema identification	208
B.2.3	product_property_definition_schema identification	208
B.2.4	product_property_representation_schema identification	209
B.2.5	management_resources_schema identification	209
B.2.6	document_schema identification	209
B.2.7	action_schema identification	209
B.2.8	certification_schema identification	209
B.2.9	approval_schema identification	210
B.2.10	contract_schema identification	210
B.2.11	security_classification_schema identification	210
B.2.12	person_organization_schema identification	210

B.2.13	date_time_schema identification	210
B.2.14	group_schema identification	211
B.2.15	effectivity_schema identification	211
B.2.16	external_reference_schema identification	211
B.2.17	support_resource_schema identification	211
B.2.18	measure_schema identification	211
B.2.19	basic_attribute_schema identification	212
Annex C (informative) Computer-interpretable listing		213
Annex D (informative) EXPRESS-G diagrams		214
Annex E (informative) Technical discussions		259
E.1	Generic product description resource structure	259
E.2	Function template for cycle detection: acyclic_object_relationship	259
E.3	Relationship template: object_relationship	260
E.4	Constraining entity instances of the basic_attribute_schema	261
Annex F (informative) Examples		262
F.1	Use of the product_definition_schema	262
F.2	Document as product	263
F.2.1	Identification of a document	263
F.2.2	Identification of a version of a document	263
F.2.3	Identification of a definition of document	264
F.2.4	Assembly structure of a document	264
F.2.5	Association of documentation to other data	264
F.2.6	Enabling use of document specific resources	264
F.2.7	Properties of a document	265
F.3	Use of the generic management resource constructs	265
F.4	Use of the measure_schema	266
F.4.1	Derived SI units	266
F.4.2	Currency conversion	268
F.4.3	Context dependent unit	269
F.4.4	Unit conversion based on an algebraic expression	269
F.4.5	Derivation of area unit and volume unit	270
F.4.6	Use of global_unit_assigned_context	271
F.5	Use of the person_organization_schema	271
F.5.1	Address of a person in an organization	271
F.5.2	Use of person_assignment	272
Bibliography		273
Index		274

Figures

Figure 1	- The relationship of the schemas of this part to the ISO10303 integration architecture	xvi
Figure D.1	- EXPRESS-G diagram of the application_context_schema (1 of 1)	215
Figure D.2	- EXPRESS-G diagram of the product_definition_schema (1 of 2)	216
Figure D.3	- EXPRESS-G diagram of the product_definition_schema (2 of 2)	217
Figure D.4	- EXPRESS-G diagram of the product_property_definition_schema (1 of 3)	218
Figure D.5	- EXPRESS-G diagram of the product_property_definition_schema (2 of 3)	219
Figure D.6	- EXPRESS-G diagram of the product_property_definition_schema (3 of 3)	220

Figure D.7 - EXPRESS-G diagram of the product_property_representation_schema (1 of 2)	221
Figure D.8 - EXPRESS-G diagram of the product_property_representation_schema (2 of 2)	222
Figure D.9 - EXPRESS-G diagram of the management_resources_schema (1 of 8)	223
Figure D.10 - EXPRESS-G diagram of the management_resources_schema (2 of 8)	224
Figure D.11 - EXPRESS-G diagram of the management_resources_schema (3 of 8)	225
Figure D.12 - EXPRESS-G diagram of the management_resources_schema (4 of 8)	226
Figure D.13 - EXPRESS-G diagram of the management_resources_schema (5 of 8)	227
Figure D.14 - EXPRESS-G diagram of the management_resources_schema (6 of 8)	228
Figure D.15 - EXPRESS-G diagram of the management_resources_schema (7 of 8)	229
Figure D.16 - EXPRESS-G diagram of the management_resources_schema (8 of 8)	230
Figure D.17 - EXPRESS-G diagram of the document_schema (1 of 1)	231
Figure D.18 - EXPRESS-G diagram of the action_schema (1 of 2)	232
Figure D.19 - EXPRESS-G diagram of the action_schema (2 of 2)	233
Figure D.20 - EXPRESS-G diagram of the certification_schema (1 of 1)	234
Figure D.21 - EXPRESS-G diagram of the approval_schema (1 of 1)	235
Figure D.22 - EXPRESS-G diagram of the contract_schema (1 of 1)	236
Figure D.23 - EXPRESS-G diagram of the security_classification_schema (1 of 1)	237
Figure D.24 - EXPRESS-G diagram of the person_organization_schema (1 of 3)	238
Figure D.25 - EXPRESS-G diagram of the person_organization_schema (2 of 3)	239
Figure D.26 - EXPRESS-G diagram of the person_organization_schema (3 of 3)	240
Figure D.27 - EXPRESS-G diagram of the date_time_schema (1 of 3)	241
Figure D.28 - EXPRESS-G diagram of the date_time_schema (2 of 3)	242
Figure D.29 - EXPRESS-G diagram of the date_time_schema (3 of 3)	243
Figure D.30 - EXPRESS-G diagram of the group_schema (1 of 1)	244
Figure D.31 - EXPRESS-G diagram of the effectivity_schema (1 of 1)	245
Figure D.32 - EXPRESS-G diagram of the external_reference_schema (1 of 1)	246
Figure D.33 - EXPRESS-G diagram of the support_resource_schema (1 of 1)	247
Figure D.34 - EXPRESS-G diagram of the measure_schema (1 of 6)	248
Figure D.35 - EXPRESS-G diagram of the measure_schema (2 of 6)	249
Figure D.36 - EXPRESS-G diagram of the measure_schema (3 of 6)	250
Figure D.37 - EXPRESS-G diagram of the measure_schema (4 of 6)	251
Figure D.38 - EXPRESS-G diagram of the measure_schema (5 of 6)	252
Figure D.39 - EXPRESS-G diagram of the measure_schema (6 of 6)	253
Figure D.40 - EXPRESS-G diagram of the basic_attribute_schema (1 of 5)	254
Figure D.41 - EXPRESS-G diagram of the basic_attribute_schema (2 of 5)	255
Figure D.42 - EXPRESS-G diagram of the basic_attribute_schema (3 of 5)	256
Figure D.43 - EXPRESS-G diagram of the basic_attribute_schema (4 of 5)	257
Figure D.44 - EXPRESS-G diagram of the basic_attribute_schema (5 of 5)	258

Tables

Table A.1 - Short names of entities	202
---	-----

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 3.

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 part of ISO 10303 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

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

This second edition of ISO 10303-41 constitutes a technical revision of the first edition (ISO 10303-41:1994), which is provisionally retained to support continued use and maintenance of implementations based on the first edition, and to satisfy the normative references of other parts of ISO 10303. This edition incorporates the corrections published in ISO 10303-41/Cor.1:1999.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. A complete list of parts of ISO 10303 is available from Internet:

<<http://www.nist.gov/sc4/editing/step/titles/>>.

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

Annexes A and B form a normative part of this part of ISO 10303. Annexes C to F are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This part of ISO 10303 is a member of the integrated resource series. Major subdivisions of this part of ISO 10303 are:

a) generic product description resources;

- 1) **application_context_schema;**
- 2) **product_definition_schema;**
- 3) **product_property_definition_schema;**
- 4) **product_property_representation_schema.**

b) generic management resources;

- 1) **management_resources_schema.**

c) support resources.

- 1) **action_schema;**
- 2) **approval_schema;**
- 3) **certification_schema;**
- 4) **contract_schema;**
- 5) **date_time_schema;**
- 6) **document_schema;**
- 7) **effectivity_schema;**
- 8) **external_reference_schema;**
- 9) **group_schema;**
- 10) **measure_schema;**
- 11) **person_organization_schema;**
- 12) **security_classification_schema;**
- 13) **support_resource_schema;**

14) **basic_attribute_schema.**

The groupings of resource schemas into these major subdivisions are shown in Figure 1. In addition, Figure 1 shows the relationship of the schemas in this part of ISO 10303 to other schemas that belong to the integrated resources of this International Standard using the EXPRESS-G notation. EXPRESS-G is defined in annex D of ISO 10303-11. The schemas illustrated in Figure 1 are components of the integrated resources.

The generic product description resources provide an overall organization for the integrated resources that are documented in other parts of ISO 10303. They specify resource constructs that provide consistent representation of facts about products in different application-specific views.

The generic management resources support the description of information that is used to manage and control product data. Together, the generic product description resources and the generic management resources are the foundations upon which application interpreted models, the normative conceptual schemas of application protocols, are built. Application interpreted models specialize selected generic management resources to elements of the integrated product description resources to satisfy the requirements that are specified in the application reference model.

The support resources are a set of shared resource constructs that are used by the ISO 10303 integrated resources. They provide an underlying consistency across the resources of ISO 10303.

This edition incorporates modifications that are upwardly compatible with the previous edition. These modifications to the EXPRESS specifications have been done so that:

- instances encoded according to ISO 10303-21 [2] 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 [3] 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.

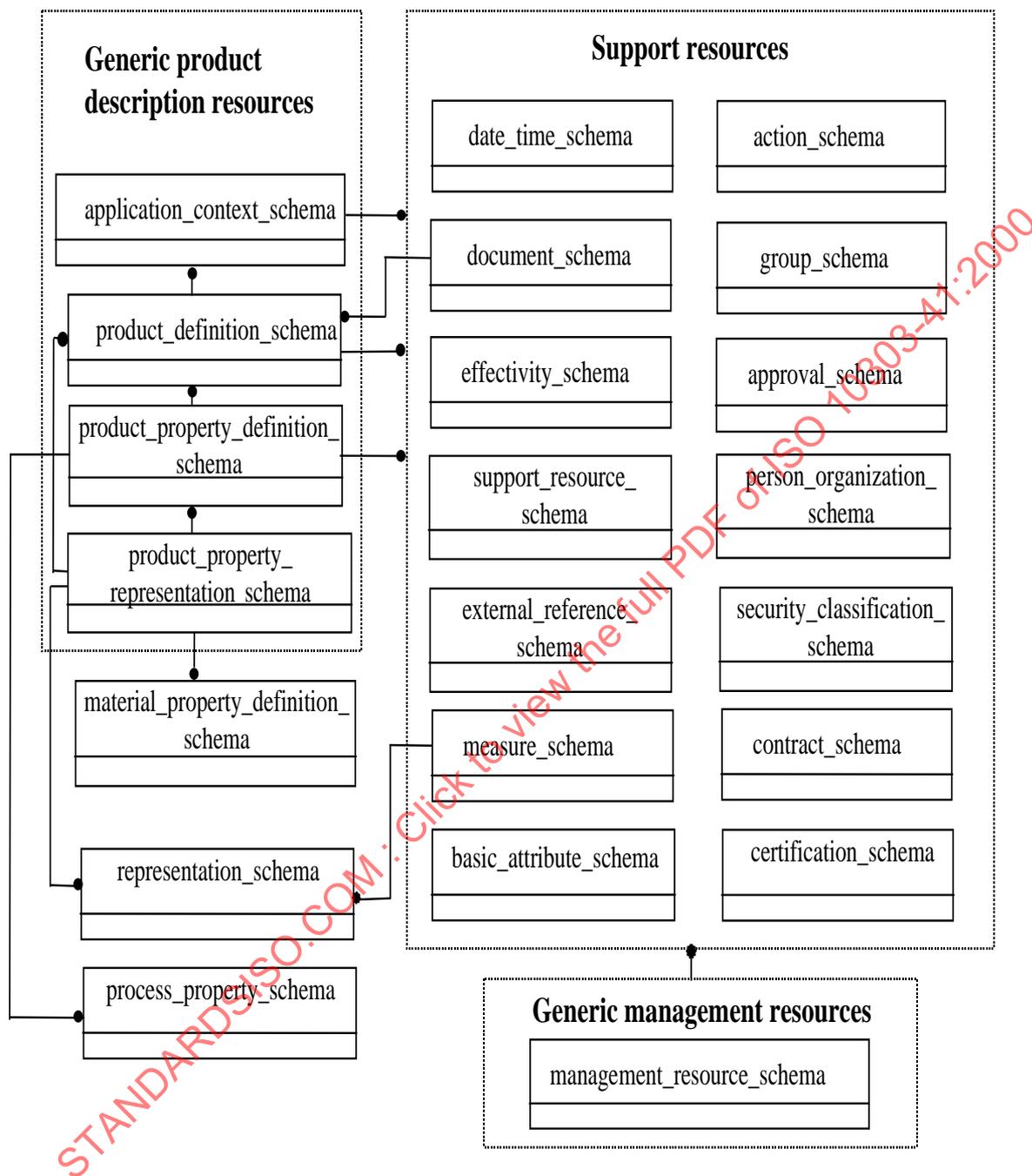


Figure 1 - The relationship of the schemas of this part to the ISO10303 integration architecture

Technical modifications to ISO 10303-41:1994 are categorized as follows: changes to the EXPRESS declarations, new EXPRESS declarations, and changes to definitions of EXPRESS entity data types.

The following EXPRESS declarations have been modified:

- **action;**
- **action_assignment;**
- **action_directive;**
- **action_method;**
- **action_method_relationship;**
- **action_relationship;**
- **action_request_assignment;**
- **action_request_solution;**
- **action_resource;**
- **action_resource_relationship;**
- **acyclic_action_method_relationship;**
- **acyclic_action_relationship;**
- **acyclic_action_resource_relationship;**
- **acyclic_approval_relationship;**
- **acyclic_document_relationship;**
- **acyclic_external_source_relationship;**
- **acyclic_group_relationship;**
- **acyclic_organization_relationship;**
- **acyclic_product_category_relationship;**
- **acyclic_product_definition_formation_relationship;**
- **acyclic_product_definition_relationship;**
- **acyclic_shape_aspect_relationship;**
- **ahead_or_behind;**
- **application_context;**

- approval_assignment;
- approval_date_time;
- approval_relationship;
- approval_role;
- bag_to_set;
- certification_assignment;
- characterized_object;
- context_dependent_shape_representation;
- contract_assignment;
- coordinated_universal_time_offset;
- date_role;
- date_time_role;
- dated_effectivity;
- day_in_month_number;
- day_in_year_number;
- derived_unit;
- dimensions_for_si_unit;
- document;
- document_reference;
- document_relationship;
- effectivity;
- effectivity_assignment;
- external_referent_assignment;
- external_source;
- external_source_relationship;
- group;

- **group_assignment;**
- **group_relationship;**
- **name_assignment;**
- **organization;**
- **organization_relationship;**
- **organization_role;**
- **organizational_address;**
- **organizational_project;**
- **person;**
- **person_and_organization;**
- **person_and_organization_role;**
- **person_role;**
- **personal_address;**
- **product;**
- **product_category_relationship;**
- **product_definition;**
- **product_definition_relationship;**
- **product_definition_substitute;**
- **product_definition_shape;**
- **product_definition_formation_relationship;**
- **product_definition_effectivity;**
- **product_definition_formation;**
- **property_definition;**
- **property_definition_representation;**
- **relatives_of_product_definitions;**
- **relatives_of_shape_representations;**
- **second_in_minute;**

- **security_classification_assignment;**
- **shape_aspect;**
- **shape_aspect_relationship;**
- **shape_definition_representation;**
- **si_unit;**
- **time_role;**
- **valid_calendar_date;**
- **versioned_action_request;**
- **week_of_year_and_day_date.**

The following EXPRESS declarations have been added:

- **action_method_assignment;**
- **action_method_role;**
- **acyclic_characterized_object_relationship;**
- **acyclic_contract_relationship;**
- **acyclic_effectivity_relationship;**
- **acyclic_event_occurrence_relationship;**
- **acyclic_externally_defined_item_relationship;**
- **acyclic_general_property_relationship;**
- **acyclic_identification_assignment_relationship;**
- **acyclic_organizational_project_relationship;**
- **acyclic_product_relationship;**
- **acyclic_time_interval_relationship;**
- **application_context_relationship;**
- **attribute_classification_assignment;**
- **attribute_type;**
- **attribute_value_assignment;**

- **attribute_value_role;**
- **celsius_temperature_measure;**
- **celsius_temperature_measure_with_unit;**
- **characterized_object_relationship;**
- **classification_assignment;**
- **classification_role;**
- **contract_relationship;**
- **date_time_or_event_occurrence;**
- **derived_property_select;**
- **description_attribute;**
- **description_attribute_select;**
- **document_production_association;**
- **document_representation_type;**
- **document_usage_constraint_assignment;**
- **document_usage_role;**
- **effectivity_context_role;**
- **effectivity_context_assignment;**
- **effectivity_relationship;**
- **event_occurrence;**
- **event_occurrence_assignment;**
- **event_occurrence_context_role;**
- **event_occurrence_context_assignment;**
- **event_occurrence_relationship;**
- **event_occurrence_role;**
- **external_identification_assignment;**
- **externally_defined_item_relationship;**
- **general_property;**

- **general_property_association;**
- **general_property_relationship;**
- **get_description_value;**
- **get_id_value;**
- **get_name_value;**
- **get_product_definitions;**
- **get_property_definition_representations;**
- **get_role;**
- **get_shape_aspects;**
- **id_attribute;**
- **id_attribute_select;**
- **identification_assignment_relationship;**
- **identification_assignment;**
- **identification_role;**
- **item_identified_representation_usage;**
- **name_attribute;**
- **name_attribute_select;**
- **object_role;**
- **organizational_project_role;**
- **organizational_project_assignment;**
- **organizational_project_relationship;**
- **product_definition_context_association;**
- **product_definition_context_role;**
- **product_or_formation_or_definition;**
- **product_relationship;**
- **relative_event_occurrence;**

- **represented_definition;**
- **role_association;**
- **role_select;**
- **time_interval;**
- **time_interval_assignment;**
- **time_interval_based_effectivity;**
- **time_interval_relationship;**
- **time_interval_role;**
- **time_interval_with_bounds;**
- **type_check_function.**

The definitions of the following EXPRESS entity data types have been modified:

- **action;**
- **action_directive;**
- **action_method;**
- **action_method_relationship;**
- **action_relationship;**
- **action_request_solution;**
- **action_request_status;**
- **action_resource;**
- **action_resource_relationship;**
- **action_resource_type;**
- **action_status;**
- **address;**
- **amount_of_substance_measure_with_unit;**
- **amount_of_substance_unit;**
- **application_context_element;**
- **application_context;**

- **application_protocol_definition;**
- **approval;**
- **approval_date_time;**
- **approval_relationship;**
- **approval_role;**
- **approval_status;**
- **area_measure_with_unit;**
- **area_unit;**
- **calendar_date;**
- **certification;**
- **certification_type;**
- **characterized_definition;**
- **characterized_object;**
- **characterized_product_definition;**
- **context_dependent_measure;**
- **context_dependent_shape_representation;**
- **context_dependent_unit;**
- **contract_type;**
- **conversion_based_unit;**
- **coordinated_universal_time_offset;**
- **date_role;**
- **date_time_role;**
- **date_time_select;**
- **dated_effectivity;**
- **derived_unit_element;**
- **descriptive_measure;**

- **dimensional_exponents;**
- **directed_action;**
- **document;**
- **document_relationship;**
- **document_type;**
- **document_with_class;**
- **effectivity;**
- **electric_current_measure_with_unit;**
- **electric_current_unit;**
- **executed_action;**
- **external_referent_assignment;**
- **external_source_relationship;**
- **externally_defined_item;**
- **global_unit_assigned_context;**
- **group_relationship;**
- **hour_in_day;**
- **identifier;**
- **label;**
- **length_measure_with_unit;**
- **length_unit;**
- **library_assignment;**
- **library_context;**
- **local_time;**
- **lot_effectivity;**
- **luminous_intensity_unit;**
- **luminous_intensity_measure_with_unit;**
- **mass_measure_with_unit;**

- **mass_unit;**
- **name_assignment;**
- **named_unit;**
- **ordinal_date;**
- **organization_relationship;**
- **organization_role;**
- **organizational_project;**
- **parameter_value;**
- **person_and_organization_role;**
- **person_organization_select;**
- **person_role;**
- **plane_angle_measure_with_unit;**
- **plane_angle_unit;**
- **pre_defined_item;**
- **product;**
- **product_category;**
- **product_category_relationship;**
- **product_concept_context;**
- **product_context;**
- **product_definition;**
- **product_definition_effectivity;**
- **product_definition_formation_relationship;**
- **product_definition_formation_with_specified_source;**
- **product_definition_relationship;**
- **product_definition_shape;**
- **product_definition_context;**

- **product_definition_substitute;**
- **property_definition_representation;**
- **property_definition;**
- **ratio_measure_with_unit;**
- **ratio_unit;**
- **security_classification;**
- **security_classification_level;**
- **serial_numbered_effectivity;**
- **shape_aspect;**
- **shape_aspect_relationship;**
- **shape_definition;**
- **shape_definition_representation;**
- **shape_representation;**
- **shape_representation_relationship;**
- **si_unit;**
- **solid_angle_measure_with_unit;**
- **solid_angle_unit;**
- **source;**
- **source_item;**
- **supported_item;**
- **text;**
- **thermodynamic_temperature_measure_with_unit;**
- **thermodynamic_temperature_unit;**
- **time_measure_with_unit;**
- **time_role;**
- **time_unit;**
- **versioned_action_request;**

- **volume_measure_with_unit**;
- **volume_unit**;
- **year_number**.

In this International Standard the same English language words may be used to refer to an object in the real world or concept, and as the name of an EXPRESS data type that represents this object or concept. The following typographical convention is used to distinguish between these. If a word or phrase occurs in the same typeface as narrative text, the referent is the object or concept. If the word or phrase occurs in a bold typeface, the referent is the EXPRESS data type. Names of EXPRESS schemas also occur in a bold typeface.

The name of an EXPRESS data type may be used to refer to the data type itself, or to an instance of the data type. The distinction between these uses is normally clear from the context. If there is a likelihood of ambiguity, either the phrase "entity data type" or "instance(s) of" is included in the text.

Double quotation marks (“ ”) denote quoted text. Single quotation marks (‘ ’) denote particular text string values.

STANDARDSISO.COM : Click to view the full PDF of 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

1 Scope

This part of ISO 10303 specifies the following:

- generic product description resources;
- generic management resources;
- support resources.

The schemas that are specified in this part of ISO 10303 are organized according to these subdivisions.

1.1 Generic product description resources

This subdivision of ISO 10303-41 specifies the resource constructs for the high level structure for the representation of products and their properties. It also specifies ISO 10303 integrated resources for the description of generic aspects of product usage, categorization of products and associations between products.

The following are within the scope of this subdivision:

- the identification of a product;
- the categorization of a product;
- the characterization of a product in one or more application contexts;
- the definition of relationships among products;
- the identification of the properties of a product or of a portion of a product;
- the identification of a representation of a property;

EXAMPLE A 3D geometric model or a table of numerical values are different representations of properties of a product.

- the description of the application context for which product data is defined.

1.2 Generic management resources

This subdivision of ISO 10303-41 specifies the resource constructs to associate administrative data with product data.

The following are within the scope of this subdivision:

- the structure for associating product data with related administrative data.

1.3 Support resources

This section of ISO 10303-41 specifies the resource constructs for administrative data, physical quantities and their units, and basic data types.

The identification and description of the following are within the scope of this subdivision:

- the identification of documents;
- the actions, action requests, and the status of actions;
- the certification, approvals, security classifications, and effectivities;
- the contracts;
- the people and organizations;
- the specification of dates and times;
- the provision of mechanisms for grouping items and referring to information that is defined outside an exchange;
- the physical quantities and their units.
- the provision of mechanisms to define values for id, name, description and role attributes.

2 Normative references

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

ISO 31 (all parts), *Quantities and units*.

ISO 1000:1992, *SI units and recommendations for the use of their multiples and of certain other units*.

ISO 8601:1988, *Data elements and interchange formats - Information interchange - Representation of dates and times*.

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

ISO 10303-44:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resource: Product structure configuration.*

ISO 10303-49:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 49: Integrated generic resources: Process structure and properties.*

3 Terms, definitions, and abbreviations

3.1 Terms defined in ISO 10303-1

For the purposes of this part of ISO 10303, the following terms and definitions given in ISO 10303-1 apply:

- application;
- application context;
- application interpreted model;
- application protocol;
- application reference model;
- data;
- information;
- integrated resource;
- information;
- product;
- product data;
- structure.

3.2 Terms defined in ISO 8601

For the purposes of this part of ISO 10303, the following terms and definitions given in ISO 8601 apply:

- calendar date;
- calendar week;
- calendar year;
- common year;
- coordinated universal time;
- day;
- Gregorian calendar;
- hour;
- leap year;
- local time;
- minute;
- month;
- ordinal date;
- second;
- week;
- year.

3.3 Other terms and definitions

For the purposes of this part of ISO 10303, the following terms and definitions apply:

3.3.1

agreement of common understanding

the result of discussions between the partners of product data exchange or sharing that ensures that all of them have the same understanding of the transferred or shared information.

NOTE The agreement may be formalized in a document.

3.3.2

annotated EXPRESS schema

an EXPRESS schema for which a natural language definition is given with each CONSTANT, TYPE, ENTITY, RULE, FUNCTION or PROCEDURE construct.

EXAMPLE Clause 5.2 of an application protocol contains an annotated EXPRESS schema.

3.3.3

class of products

the set of all products satisfying a type of product. The elements of the set are referred to as members of the class.

NOTE 1 A class need not have any members.

NOTE 2 The definition of the type will determine whether the size of the set varies with time.

NOTE 3 This definition was adapted from ISO/IEC 10746-2:1996 [1].

3.3.4

real object

an object that is existing or was existing in the real world and that can or could be uniquely identified.

NOTE An individual object may result from a manufacturing process or it may exist independently of any human process.

EXAMPLE The Moon is an individual object that is not the result of a human process.

3.3.5

type of product

a predicate characterizing a collection of products. A product is of the type, or satisfies the type, if the predicate holds for that product.

NOTE This definition was adapted from ISO/IEC 10746-2:1996 [1].

3.4 Abbreviations

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

SI International System of Units

URL Uniform Resource Locator

4 Application context

The following EXPRESS declaration begins the **application_context_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA application_context_schema;

REFERENCE FROM basic_attribute_schema           -- ISO 10303-41
  (description_attribute,
   get_description_value,
   get_id_value,
   id_attribute);

REFERENCE FROM date_time_schema                 -- ISO 10303-41
  (year_number);

REFERENCE FROM support_resource_schema         -- ISO 10303-41

```

```
(identifier,
  label,
  text);
(*
```

NOTE 1 The schemas referenced above are specified in the following part of ISO 10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
date_time_schema	clause 16 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema is one of the generic product description resource schemas. The relationships between the generic product description resource schemas are given in annex E.1.

4.1 Introduction

The subject of the **application_context_schema** is the applicable usages of product data. This schema provides a mechanism for application protocols to define a frame of reference or context that applies to particular sets of product data.

This schema also provides a means to identify application protocols.

4.2 Fundamental concepts and assumptions

The meaningful exchange of product data requires the identification of the application context in which that product data is defined. An application context may have several context elements. Each context element may be referenced by a different set of product data.

4.3 Application context entity definitions

4.3.1 application_context

An **application_context** is a context in which product data is defined and has meaning. An **application_context** represents various types of information that relate to product data and may affect the meaning and usage of that data.

EXPRESS specification

```
*)
ENTITY application_context;
  application      : label;
DERIVE
  description      : text := get_description_value (SELF);
  id               : identifier := get_id_value (SELF);
INVERSE
  context_elements : SET [1:?] OF application_context_element
                    FOR frame_of_reference;
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
```

```

        'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
        'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- application_context
(*

```

Attribute definitions

application: the **label** by which the **application_context** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **application_context**.

NOTE 2 The description attribute aids in defining the usage of product data in an **application_context**.

NOTE 3 This attribute is an upwardly compatible addition to **application_context** as specified in ISO 10303-41:1994.

id: the **identifier** that distinguishes the **application_context**.

NOTE 4 This attribute is an upwardly compatible addition to **application_context** as specified in ISO 10303-41:1994.

NOTE 5 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

context_elements: the set of instances of the entity data type **application_context_element** that defines the various aspects of the **application_context**.

Formal propositions:

WR1: Each **application_context** shall be the described_item in at most one **description_attribute**.

NOTE 6 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

WR2: Each **application_context** shall be the identified_item in at most one **id_attribute**.

NOTE 7 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 8 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

4.3.2 application_context_element

An **application_context_element** is an aspect of an application context in which product data is defined. This aspect contains information about the conditions in which the product data exists.

EXPRESS specification

```

*)
ENTITY application_context_element
  SUPERTYPE OF (ONEOF (library_context,
                      product_concept_context,

```

```

        product_context,
        product_definition_context));
    name          : label;
    frame_of_reference : application_context;
END_ENTITY; -- application_context_element
(*)

```

Attribute definitions

name: the **label** by which the **application_context_element** is known.

EXAMPLE 'Functional definition', 'physical definition', and 'usage occurrence' are examples of names.

frame_of_reference: the **application_context** of which this **application_context_element** is a member.

4.3.3 application_context_relationship

An **application_context_relationship** relates two instances of **application_context** and provides a description of their relationship.

NOTE 1 The role of **application_context_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Without any further specialization, this relationship does not imply that the product data associated with **relating_context** are also associated with the **related_context**.

NOTE 3 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 4 This entity, together with the **application_context** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```

*)
ENTITY application_context_relationship;
    name          : label;
    description    : OPTIONAL text;
    relating_context : application_context;
    related_context : application_context;
END_ENTITY; -- application_context_relationship;
(*)

```

Attribute definitions

name: the **label** by which the **application_context_relationship** is known.

description: the **text** that characterizes the **application_context_relationship**. The value of the attribute need not be specified.

relating_context: one of the instances of **application_context** that is a part of the relationship.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_context: the other instance of **application_context** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 6 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

4.3.4 application_protocol_definition

An **application_protocol_definition** is the identification of an application protocol.

EXPRESS specification

```
*)
ENTITY application_protocol_definition;
    status : label;
    application_interpreted_model_schema_name : label;
    application_protocol_year : year_number;
    application : application_context;
END_ENTITY; -- application_protocol_definition;
(*
```

Attribute definitions

status: the label that provides a user interpretable designation for the level of completion of the action.

application_interpreted_model_schema_name: the EXPRESS schema name of the application interpreted model.

application_protocol_year: the year when the application protocol attained the status given by the status attribute.

application: the application context of the application protocol.

4.3.5 library_context

The **library_context** is a type of **application_context_element** that represents information about a library, and defines a context for elements of the library. This information applies to product data that is related to a library through use of the **library_assignment** entity data type (see 8.4.29).

EXPRESS specification

```
*)
ENTITY library_context
    SUBTYPE OF (application_context_element);
    library_reference : label;
END_ENTITY; -- library_context
(*
```

Attribute definitions

SELF\application_context_element.name: the **label** that identifies or characterizes the type of library.

NOTE The name attribute can represent the nature of the source of the library, the nature of the contents of the library, or both.

EXAMPLE 1 'Dewey Decimal', 'ISO 13584 library', and 'supplier catalogue' are examples of type of library.

library_reference: the **label** that identifies the library.

EXAMPLE 2 'ACME self-propelled rocket sleds catalogue' is an example of a library name.

4.3.6 product_concept_context

A **product_concept_context** is a type of **application_context_element** that defines a context for a **product_concept** (see ISO 10303-44).

A **product_concept_context** represents information that relates to the characterization of potential purchasers of a product. Such information may affect the meaning and usage of the product data.

EXPRESS specification

```
*)
ENTITY product_concept_context
  SUBTYPE OF (application_context_element);
  market_segment_type : label;
END_ENTITY; -- product_concept_context
(*
```

Attribute definitions

market_segment_type: the **label** that identifies the kind of consumer preferences associated with a product.

EXAMPLE 'Luxury automobiles', 'laptop personal computers', and 'budget personal stereos' are examples of market_segment_types.

4.3.7 product_context

The **product_context** is a type of **application_context_element** that represents life cycle independent information about a product (see 5.4.1). This information describes the discipline in which data about the product are created or used.

NOTE 1 Different annotated EXPRESS schemas that use or specialize this entity data type can make use of common values for the attributes of the **product_context** entity data type in order to establish consistency among schemas that relate to the same types or kinds of products.

EXAMPLE 1 Two annotated EXPRESS schemas that specify product data structures for design and construction of buildings respectively can each constrain the values of the name attribute to be 'building', and the discipline_type attribute to be 'architecture'.

EXPRESS specification

```
*)
ENTITY product_context
  SUBTYPE OF (application_context_element);
  discipline_type : label;
END_ENTITY; -- product_context
(*
```

Attribute definitions

SELF\application_context_element.name: the **label** that identifies a particular industry, subject area, or kind of product for which product data is created or used.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 2 'ship', 'aircraft' and 'building' are examples of names of product contexts.

discipline_type: the **label** that identifies a particular field of practice involved in creating or using product data.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 3 'electronics', 'engineering', and 'architecture' are examples of discipline types.

4.3.8 product_definition_context

The **product_definition_context** is a type of **application_context_element** that represents information about the stage in the product life cycle for which a **product_definition** (see 5.4.4) is created or used.

NOTE 1 Different annotated EXPRESS schemas that use or specialize this entity data type can make use of common values for the attributes of the **product_definition_context** entity data type in order to maintain consistency between schemas that relate to the same or related life-cycle phases.

EXAMPLE 1 Two annotated EXPRESS schemas that specify product data structures for conceptual design and detailed design of process plants respectively can each constrain the values of the name attribute to be 'conceptual design' and 'detailed design', and the **life_cycle_stage** attribute to be 'design'.

EXPRESS specification

```
* )
ENTITY product_definition_context
  SUBTYPE OF (application_context_element);
  life_cycle_stage : label;
END_ENTITY; -- product_definition_context
(*
```

Attribute definitions

SELF\application_context_element.name: the **label** that identifies a subdivision or characterization of a life cycle stage for which product data is created or used.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 2 'required' and 'planned' are examples of names of product definition contexts, used in conjunction with a life cycle stage of 'thermal analysis'.

life_cycle_stage: the **label** that identifies a stage in the life cycle of a product.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 3 'preliminary design' and 'manufacturing planning' are examples of life cycle stages.

EXPRESS specification

```
*)
END_SCHEMA; -- application_context_schema
(*
```

5 Product definition

The following EXPRESS declaration begins the **product_definition_schema** and identifies the necessary external references.

EXPRESS specification

```
*)
SCHEMA product_definition_schema;

REFERENCE FROM application_context_schema -- ISO 10303-41
  (product_context,
   product_definition_context);

REFERENCE FROM basic_attribute_schema -- ISO 10303-41
  (get_id_value,
   get_name_value,
   name_attribute);

REFERENCE FROM document_schema -- ISO 10303-41
  (document);

REFERENCE FROM effectivity_schema -- ISO 10303-41
  (effectivity);

REFERENCE FROM support_resource_schema -- ISO 10303-41
  (bag_to_set,
   identifier,
   label,
   text);
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

application_context_schema	clause 4 of this part of ISO 10303
basic_attribute_schema	clause 22 of this part of ISO 10303
document_schema	clause 9 of this part of ISO 10303
effectivity_schema	clause 18 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema is one of the generic product description resource schemas. The relationships between the generic product description resource schemas are given in annex E.1.

5.1 Introduction

The subject of the **product_definition_schema** is the identification of products, the categorization of products, definitions of products, and the relationships among them. This schema provides for the generic aspects of product definition.

EXAMPLE The identification of products and definitions of products, the grouping of products according to classification schemes and the definition of various kinds of relationship between products are all examples of generic aspects of product definition.

5.2 Fundamental concepts and assumptions

The use of the entity data types of this schema to represent products is specified in the viewpoint as defined by the requirements of an annotated EXPRESS schema that uses or specializes this part of ISO 10303. Every product need not be represented by an instance of the entity data type **product**. Some products may be represented by other constructs in the integrated resources.

EXAMPLE 1 The constructs **product_definition_formation** and **product_definition**, defined in the **product_definition_schema** of this part of ISO 10303, are examples of other entity data types that may represent a product.

EXAMPLE 2 The construct **action_resource**, defined in ISO 10303-49, is an example of an entity data type that may represent a product.

A single product may have multiple groups of definitions associated with it; each group is valid in a given application context.

EXAMPLE 3 An application context may be the manufacture of ball-point pens. A particular design of ball-point pen has multiple versions. Each version is described through a group of product definitions. An older version has a solid cap whereas a more recent version has a hole in the cap to prevent people from choking if they swallow it. The characteristics of the two versions would be different from each other because the later version would have a hole in the cap whereas the earlier version would not. This schema could be used to define the ball-point pen with a solid cap as one product and the one with a hole in the cap as another. On the other hand it could also be used to define both pens as two versions of a single product. The approach taken would depend upon the requirements of the application context.

The definition of a product may include the properties that characterize it.

EXAMPLE 4 An integrated circuit product will have a functional definition represented by a circuit schematic diagram and a physical definition represented by a circuit layout diagram.

Definitions of products may be related to each other in various ways. This part of ISO 10303 provides the resources needed to identify the relationships among products and to be able to characterize those relationships.

EXAMPLE 5 The ball-point pen cap would be related in an assembly relationship to the ball-point pen itself and one of the caps could be substituted for the other in some circumstances.

There are two ways to represent the concept of real object.

— a real object can be represented by an instance of **product** if the subject of an annotated Express schema is concerned with real objects or if it is needed to describe a real object at various life-cycle stages and in various configurations; in this case, if a real object is identified by a serial number, this serial number can be stored in the **id** attribute of the **product**;

EXAMPLE 6 An airplane, in service during 30 years, may have different configurations during its life.

— a real object can be represented by an instance of **product_definition** if the subject of an annotated EXPRESS schema is primarily concerned with the design phase of products and if the concept of real object is considered as a particular stage in the design phase of a product; in this case, if a real object is identified by a serial number, this serial_number can be stored in the id attribute of the **product_definition**.

EXAMPLE 7 A prototype may be considered as a particular definition of a product.

Product definition facts are independent of properties. The ways in which these facts may be combined are prescribed by the relationships that are defined in this schema. Each product definition fact can be interpreted in any application context; the ways in which these facts are to be interpreted are prescribed in annotated EXPRESS schemas that use or specialise the resources defined in this part of ISO 10303.

NOTE An example of the way in which this schema may be used is given in annex F.1.

5.3 Product definition type definition: source

The **source** type is a list of alternate choices that enables one to characterize whether a product is to be manufactured within an organization, bought, or the fact that this information is not known.

EXPRESS specification

```
* )
TYPE source = ENUMERATION OF
  (made,
   bought,
   not_known);
END_TYPE; -- source
(*
```

5.4 Product definition entity definitions

5.4.1 product

A **product** represents a product or a type of product (see 3.3.4).

NOTE 1 The term product is defined in ISO 10303-1.

EXAMPLE 1 The SS Titanic is a product that could be represented by **product**.

EXAMPLE 2 Lifeboat is a class of products that could be represented by the entity data type **product**. Each lifeboat on the SS Titanic is a member of this class.

A **product** depends on one or more instances of **product_context** specifying a frame of reference that determines the validity of the information held about the product or class of products.

NOTE 2 Products that this entity data type can represent include:

- products existing in the real world;
- products that may come into existence as a consequence of some realization process;
- products that are functions;
- products that require further detail so they may serve as the basis for other physically realizable

products.

EXAMPLE 3 The computer used to edit this document, the Eiffel Tower, and coal are examples of products existing in the real world.

EXAMPLE 4 A company develops a new lawnmower, a product that will be realized through a manufacturing and assembly process.

EXAMPLE 5 Heating, lighting, and Internet connectivity are examples of functions.

EXAMPLE 6 A company may develop a generic engine that is described by a schematic drawing showing its main principles and components. This product may be used as the basis for a range of engines that share characteristics such as multi-valve and operating temperature. Then, when defining a new engine, designers create a specialization of this generic engine with specific characteristics such as cylinder volume and power.

NOTE 3 The products or classes of products that are represented by **product** are specified in annotated EXPRESS schemas that use or specialize this entity data type.

EXPRESS specification

```
*)
ENTITY product;
  id           : identifier;
  name        : label;
  description  : OPTIONAL text;
  frame_of_reference : SET [1:?] OF product_context;
END_ENTITY; -- product
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **product**.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 5 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

EXAMPLE 7 Part numbers, stock item numbers, and serial numbers are examples of product identifiers.

name: the **label** by which the **product** is known.

EXAMPLE 8 'Widget' is an example of name.

description: the **text** that characterizes the **product**. The value of the attribute need not be specified.

frame_of_reference: a set of **product_context** entities that defines the contexts within which the data associated with the **product** is relevant.

5.4.2 product_category

A **product_category** identifies a type of product

EXAMPLE In an application protocol whose context includes manufactured parts, 'mechanical part', 'electrical part', 'structural part', 'piping part', 'water pipe', and 'hot water pipe' are examples of types of product

EXPRESS specification

```
*)
ENTITY product_category;
  name      : label;
  description : OPTIONAL text;
DERIVE
  id          : identifier := get_id_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- product_category
(*
```

Attribute definitions

name: the **label** by which the **product_category** is known.

description: the **text** that characterizes the **product_category**. The value of the attribute need not be specified.

id: the **identifier** that distinguishes the **product_category**. The value of this attribute need not be specified.

NOTE 1 This attribute is an upwardly compatible addition to **product_category** as specified in ISO 10303-41:1994.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 3 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

Formal propositions:

WR1: Each **product_category** shall be the **identified_item** in at most one **id_attribute**.

NOTE 4 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

Informal propositions:

IP1: If a **product_category** participates in a **product_category_relationship**, the value of the name attribute shall not be identical to the name attribute within any of its parent **product_category** entity data types.

NOTE 5 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

5.4.3 product_category_relationship

A **product_category_relationship** hierarchically relates one category with another and provides a description of their relationship.

NOTE 1 The role of **product_category_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing

this information.

EXAMPLE 1 The two sub-level **product_category**s called 'cold water pipe' and 'hot water pipe' can be related to each other through the more generic **product_category** called 'piping part' by using two **product_category_relationship**s, one **product_category_relationship** between 'piping part' and 'cold water pipe' and the other **product_category_relationship** between 'piping part' and 'hot water pipe'.

If a **product** belongs to a **product_category**, it also belongs to all of the parent categories.

NOTE 2 Networks of instances of **product_category** may be defined using this entity.

EXAMPLE 2 A **product_category** called 'piping part' may be the parent category of sub-categories called 'cold water pipe' and 'hot water pipe'. The **product_category** called 'hot water pipe' may be a sub-category of another **product_category** called 'boiler output pipe'. This is an example of a **product_category** network because the **product_category** called 'hot water pipe' has two parents.

NOTE 3 This entity, in conjunction with the **product_category** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
* )
ENTITY product_category_relationship;
  name      : label;
  description : OPTIONAL text;
  category  : product_category;
  sub_category : product_category;
WHERE
  WR1: acyclic_product_category_relationship
      (SELF, [SELF.sub_category]);
END_ENTITY; -- product_category_relationship
( *
```

Attribute definitions

name: the **label** by which the **product_category_relationship** is known.

description: the **text** that characterizes the **product_category_relationship**. The value of the attribute need not be specified.

category: the parent of the sub_category.

EXAMPLE 3 In the previous example, 'piping part' would be the category for 'cold water pipe' and 'hot water pipe'.

sub_category: a child of the category.

EXAMPLE 4 In the previous example 'cold water pipe' would be the sub_category for 'piping part' in one case and 'hot water pipe' would be the sub_category in the other.

Formal propositions:

WR1: A graph of **product_category** instances shall not be cyclic.

5.4.4 product_definition

The **product_definition** entity data type represents an aspect of a product, or of a class of products, for an identified life cycle stage. The life cycle stage for which a **product_definition** exists may be further characterized by discipline, by usage, or by both.

NOTE 1 The **product_definition** entity type supports the representation of different views of a product for different purposes. Multiple views of the same product, or class of products, are represented by different instances of **product_definition** for the same **product_definition_formation**.

EXAMPLE 1 The design of the SS Titanic and the as-built description of the SS Titanic can be represented as two instances of **product_definition** for the product that represents the ship itself.

The **product_definition** entity data type may represent particular products that are the members of an identified class of products.

EXAMPLE 2 Each individual lifeboat on the SS Titanic can be represented by an instance of **product_definition**, in which the associated **product** represents the class of products whose members are the lifeboats.

NOTE 2 A **product_definition** can identify an occurrence of a product.

EXAMPLE 3 The left front wheel of a car can be identified by an instance of **product_definition** in the context of the car.

The **product_definition** entity data type acts as an aggregator for information about the properties of products.

EXAMPLE 4 The designed shape of an aeroplane can be identified by an instance of the **product_definition_shape** entity data type (see 6.4.6) that is associated with the design **product_definition** of the aeroplane.

The usage of a **product_definition** in another context is specified through its participation in a **product_definition_relationship** as the related_product_definition in which the using context is specified by the **frame_of_reference** of the **relating_product_definition**.

If a **product_definition** is considered in multiple contexts, the **product_definition_context_association** shall be used to specify a collection of **product_definition_contexts**.

EXPRESS specification

```

*)
ENTITY product_definition;
  id                : identifier;
  description       : OPTIONAL text;
  formation         : product_definition_formation;
  frame_of_reference : product_definition_context;
DERIVE
  name              : label := get_name_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- product_definition
(*)

```

Attribute definitions

id: the **identifier** that distinguishes the **product_definition**.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 4 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

description: the **text** that characterizes the **product_definition**. The value of the attribute need not be specified.

formation: the **product_definition_formation** to which the **product_definition** relates.

frame_of_reference: the **product_definition_context** in which the **product_definition** is defined.

name: the **label** by which the **product_definition** is known.

NOTE 5 This attribute is an upwardly compatible addition to **product_definition** as specified in ISO 10303-41:1994.

NOTE 6 The name_attribute data type is defined in clause 22 of this part of ISO 10303.

Formal propositions:

WR1: Each **product_definition** shall be the named_item in at most one name_attribute.

NOTE 7 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

5.4.5 product_definition_context_association

A **product_definition_context_association** associates a **product_definition_context** with a **product_definition**, and specifies the meaning of this association.

NOTE 1 **Product_definition_context_association** may be used to associate a using context to a **product_definition**. The defining context is associated to the **product_definition** using product_definition.frame_of_reference.

EXAMPLE A **product_definition**, initially defined in the context of 'mechanical design', and in which the representation of the shape of a product is defined, may also be relevant in the context of 'process-planning design'. For such a case, the **product_definition_context_association** would relate the **product_definition** with the **product_definition_context** characterizing the context 'process_planning design' and the name of the role attribute would be 'other relevant context'.

EXPRESS specification

```
* )
ENTITY product_definition_context_association;
  definition          : product_definition;
  frame_of_reference : product_definition_context;
  role                : product_definition_context_role;
END_ENTITY; -- product_definition_context_association
```

(*

Attribute definitions

definition: the reference to a **product_definition** associated to a context.

frame_of_reference: the **product_definition_context** that defines the stage in the product life cycle with which the **product_definition** is to be associated.

role: the **product_definition_context_role** that specifies the purpose of the association of the context of this **frame_of_reference** attribute with the **product_definition** of the definition attribute.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

5.4.6 product_definition_context_role

A **product_definition_context_role** describes the purpose of a **product_definition_context**.

EXPRESS specification

```
*)
ENTITY product_definition_context_role;
  name      : label;
  description : OPTIONAL text ;
END_ENTITY; -- product_definition_context_role
(*
```

Attribute definitions

name: the **label** by which the **product_definition_context_role** is known.

description: the **text** that characterizes the **product_definition_context_role**. The value of the attribute need not be specified.

5.4.7 product_definition_effectivity

A **product_definition_effectivity** is the identification of a valid use of a particular **product_definition** in the context of its participation in a given **product_definition_relationship**. The referenced **product_definition** is the related_product_definition attribute of the **product_definition_relationship**.

NOTE 1 This entity data type is kept in this edition in order to ensure upward compatibility. The entity data type **effectivity_assignment**, introduced in this edition of this part of ISO 10303, provides extended capabilities.

NOTE 2 Appropriate subtypes of **product_definition_relationship** and further information about the way to describe assemblies are specified in ISO 10303-44.

EXPRESS specification

```
*)
ENTITY product_definition_effectivity
  SUBTYPE OF (effectivity);
  usage : product_definition_relationship;
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'MANAGEMENT_RESOURCES_SCHEMA.' +
    'EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY')) = 0;
```

```
END_ENTITY; -- product_definition_effectivity
(*
```

Attribute definitions

usage: the **product_definition_relationship** that defines the context of the **effectivity**.

Formal propositions:

WR1: A **product_definition_effectivity** shall not be referred to as **effectivity_assignment.assigned_effectivity**.

5.4.8 product_definition_formation

A **product_definition_formation** is a collector of definitions of a product.

EXAMPLE 1 An application interpreted model could use this entity to support the identification of different versions of a single **product**. Each version would be described by a group of instances of the entity data type **product_definition_formation** and each group, identified by a **product_definition_formation**, would be associated with the same **product**.

EXPRESS specification

```
*)
ENTITY product_definition_formation;
  id          : identifier;
  description : OPTIONAL text;
  of_product  : product;
UNIQUE
  UR1: id, of_product;
END_ENTITY; -- product_definition_formation
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **product_definition_formation**.

NOTE 1 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

EXAMPLE 2 Part version number is an example of a **product_definition_formation** identifier.

description: the **text** that characterizes the **product_definition_formation**. The value of the attribute need not be specified.

NOTE 2 This attribute may be used to depict differences in the purpose and function of different formations of a single **product**.

of_product: the **product** to which the **product_definition_formation** belongs.

NOTE 3 A **product** is associated with one or more **product_definition_formation** instances through the implicit inverse of this relationship.

Formal propositions:

UR1: The **id** of each **product_definition_formation** shall be unique within the collection of **product_definition_formation**s that are related to the same product (through their attribute of **_product**).

5.4.9 product_definition_formation_relationship

A **product_definition_formation_relationship** relates two instances of the entity data type **product_definition_formation** and provides an identification and description of this relationship.

NOTE 1 A relationship may exist between instances of the entity data type **product_definition_formation** that relate to different instances of the entity data type **product** or between different formations of the same **product**.

NOTE 2 The role of **product_definition_formation_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 3 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 4 This entity, together with the **product_definition_formation** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
* )
ENTITY product_definition_formation_relationship;
  id                : identifier;
  name              : label;
  description       : OPTIONAL text;
  relating_product_definition_formation : product_definition_formation;
  related_product_definition_formation : product_definition_formation;
END_ENTITY; -- product_definition_formation_relationship
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **product_definition_formation_relationship**.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 6 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **product_definition_formation_relationship** is known.

description: the **text** that characterizes the **product_definition_formation_relationship**. The value of the attribute need not be specified.

relating_product_definition_formation: one of the instances of **product_definition_formation** that is a part of the relationship.

NOTE 7 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_product_definition_formation: the other instance of **product_definition_formation** that is

a part of the relationship. If one element of the relationship is dependent up on the other, this attribute shall be the dependent one.

NOTE 8 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

5.4.10 **product_definition_formation_with_specified_source**

A **product_definition_formation_with_specified_source** is a type of **product_definition_formation** whose source is identified to be either manufactured within an organization, bought, or the fact that this information is not known.

EXPRESS specification

```
* )
ENTITY product_definition_formation_with_specified_source
  SUBTYPE OF (product_definition_formation);
  make_or_buy : source;
END_ENTITY; -- product_definition_formation_with_specified_source
(*
```

Attribute definitions

make_or_buy: the characterization of the source of the **product_definition_formation**.

5.4.11 **product_definition_relationship**

A **product_definition_relationship** relates two instances of the entity data type **product_definition** and provides an identification and description of this relationship.

NOTE 1 A relationship may exist between instances of the entity data type **product_definition** that relate to different instances of the entity data type **product** or between different definitions of the same **product**.

EXAMPLE 1 The relationships within a bill-of-materials structure are examples of **product_definition_relationship** entity data types that associate different **products**. The relationship between a sketch and a detailed design is an example of a **product_definition_relationship** that associates different definitions of a single **product**.

NOTE 2 A single **product_definition** may be used more than once within the description of a **product**.

EXAMPLE 2 The same component could be used more than once in the same assembly. Each usage of the component would be specified as an instance of the **product_definition_relationship** entity.

NOTE 3 The role of **product_definition_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 4 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 5 This entity, together with the **product_definition** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```

*)
ENTITY product_definition_relationship;
  id          : identifier;
  name       : label;
  description : OPTIONAL text;
  relating_product_definition : product_definition;
  related_product_definition  : product_definition;
END_ENTITY; -- product_definition_relationship
(*

```

Attribute definitions

id: the **identifier** that distinguishes the **product_definition_relationship**.

NOTE 6 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 7 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **product_definition_relationship** is known.

description: the **text** that characterizes the **product_definition_relationship**. The value of the attribute need not be specified.

relating_product_definition: one of the instances of **product_definition** that is a part of the relationship.

NOTE 8 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 3 If the **product_definition_relationship** is an assembly component relationship, the **relating_product_definition** may be the assembly.

related_product_definition: the other instance of **product_definition** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 9 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 4 In an assembly, the **related_product_definition** may be the **product_definition** that is an element of the assembly.

5.4.12 product_definition_substitute

A **product_definition_substitute** is an association of a **product_definition** with a **product_definition_relationship**. The association states the fact that the **product_definition** may replace, in the context of the relationship, the **product_definition** that is specified in the **related_product_definition** attribute of the **product_definition_relationship**.

NOTE 1 If a **product_definition_relationship** exists between a component part and its assembly, a **product_definition_substitute** could be used to capture an allowable substitute, that is, a different component part that may

be used as a replacement in the same assembly.

EXAMPLE 1 Two kinds of ball-point pens may be manufactured: a standard model and a deluxe model. Each model of the pen would be specified as a separate **product_definition** related to a particular kind of nib: a standard nib and a deluxe nib, respectively. The fact that the deluxe nib and the standard nib are interchangeable only in the context of the standard model of ball-point pen can be established with this construct.

EXPRESS specification

```

*)
ENTITY product_definition_substitute;
  description          : OPTIONAL text;
  context_relationship : product_definition_relationship;
  substitute_definition : product_definition;
DERIVE
  name                : label := get_name_value (SELF);
WHERE
  WR1 : context_relationship.related_product_definition :<>:
        substitute_definition;
  WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- product_definition_substitute
(*

```

Attribute definitions

description: the **text** that characterizes the **product_definition_substitute**. The value of the attribute need not be specified.

context_relationship: the **product_definition_relationship** that specifies the context in which the replacement may occur.

EXAMPLE 2 In the previous example, the relating_product_definition of a **product_definition_relationship** identified as a context_relationship would be the standard model of the ball-point pen and the related_product_definition would be the standard nib.

substitute_definition: the **product_definition** that acts as an allowed replacement for the related_product_definition of the context_relationship.

EXAMPLE 3 In the previous example, this attribute would identify the deluxe nib .

name: the **label** by which the **product_definition_substitute** is known.

NOTE 2 This attribute is an upwardly compatible addition to **product_definition** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: A **product_definition** shall not be defined as a substitute for itself.

WR2: Each **product_definition_substitute** shall be the named_item in at most one name_attribute.

NOTE 3 The **name_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

5.4.13 product_definition_with_associated_documents

A **product_definition_with_associated_documents** is a **product_definition** in which the associated documents participate in the definition of the product.

EXPRESS specification

```
*)
ENTITY product_definition_with_associated_documents
  SUBTYPE OF (product_definition);
  documentation_ids : SET[1:?] OF document;
END_ENTITY; -- product_definition_with_associated_documents
(*
```

Attribute definitions

documentation_ids: the set of instances of **document** entity data type that are associated with the **product_definition**.

5.4.14 product_related_product_category

A **product_related_product_category** is a **product_category** that identifies members of the class of products that satisfy the type identified by the category.

EXPRESS specification

```
*)
ENTITY product_related_product_category
  SUBTYPE OF (product_category);
  products : SET [1:?] OF product;
END_ENTITY; -- product_related_product_category
(*
```

Attribute definitions

products: a set of **products** that belong to the **product_related_product_category**.

5.4.15 product_relationship

A **product_relationship** relates two instances of the entity data type **product** and provides an identification and description of their relationship.

NOTE 1 The role of **product_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **product** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```

*)
ENTITY product_relationship;
  id          : identifier;
  name       : label;
  description : OPTIONAL text;
  relating_product : product;
  related_product : product;
END_ENTITY; -- product_relationship
(*)

```

Attribute definitions

id: the **identifier** that distinguishes the **product_relationship**.

NOTE 4 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **product_relationship** is known.

description: the **text** that characterizes the **product_relationship**. The value of the attribute need not be specified.

relating_product: one of the instances of **product** that is a part of the relationship.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_product: the other instance of **product** that is a part of the relationship. If one element of the relationship is dependent up on the other, this attribute shall be the dependent one.

NOTE 6 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

5.5 Product definition function definitions

5.5.1 acyclic_product_category_relationship

The **acyclic_product_category_relationship** function determines whether the graph of instances of the entity data type **product_category** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **product_category_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE The algorithm of the function is explained in annex E.2.

EXPRESS specification

```

*)
FUNCTION acyclic_product_category_relationship
  (relation : product_category_relationship;
   children : SET of product_category): BOOLEAN;

LOCAL

```

```

x          : SET OF product_category_relationship;
local_children : SET OF product_category;
END_LOCAL;

REPEAT i := 1 to HIINDEX(children);
  IF relation.category ::= children [i] THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
x := bag_to_set(USEDIN (relation.category,
                      'PRODUCT_DEFINITION_SCHEMA.' +
                      'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY'));
local_children := children + relation.category;
IF SIZEOF(x) > 0 THEN
  REPEAT i := 1 to HIINDEX(x);
    IF NOT acyclic_product_category_relationship(x[i], local_children) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
END_IF;
RETURN(TRUE);
END_FUNCTION; -- acyclic_product_category_relationship
(*)

```

Argument definitions:

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

children: (input) the set of instances of the entity data type **product_category** that the function is searching for in the category field of the **relation** argument.

5.5.2 **acyclic_product_definition_formation_relationship**

The **acyclic_product_definition_formation_relationship** function determines whether the graph of instances of the entity data type **product_definition_formation** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **product_definition_formation_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **product_definition_formation_relationship** entity include rules that use this function.

EXPRESS specification

```

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

LOCAL
  x          : SET OF product_definition_formation_relationship;
END_LOCAL;

IF relation.relativing_product_definition_formation IN relatives THEN

```

```

    RETURN (FALSE);
END_IF;
x := QUERY (pdf <* bag_to_set
            (USEDIN (relation.relying_product_definition_formation,
                    'PRODUCT_DEFINITION_SCHEMA.' +
                    'PRODUCT_DEFINITION_FORMATION_RELATIONSHIP.' +
                    'RELATED_PRODUCT_DEFINITION_FORMATION')) |
            specific_relation IN TYPEOF (pdf));
REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_product_definition_formation_relationship
        (x[i],
         relatives + relation.relying_product_definition_formation,
         specific_relation) THEN
        RETURN(FALSE);
    END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_product_definition_formation_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **product_definition_formation** that the function is searching for in the **relying_product_definition_formation** parameter of the relation argument.

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

5.5.3 **acyclic_product_definition_relationship**

The **acyclic_product_definition_relationship** function determines whether the graph of instances of the entity data type **product_definition_formation** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **product_definition_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **product_definition_relationship** entity include rules that use this function.

EXPRESS specification

```

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

LOCAL
  x          : SET OF product_definition_relationship;
END_LOCAL;

IF relation.relying_product_definition IN relatives THEN
  RETURN (FALSE);

```

```

END_IF;
x := QUERY (pd <* bag_to_set
            (USEDIN (relation.relatng_product_definition,
                    'PRODUCT_DEFINITION_SCHEMA.' +
                    'PRODUCT_DEFINITION_RELATIONSHIP.' +
                    'RELATED_PRODUCT_DEFINITION')) |
            specific_relation IN TYPEOF (pd));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_product_definition_relationship
    (x[i],
     relatives + relation.relatng_product_definition,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_product_definition_relationship
(*)

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **product_definition** that the function is searching for in the relating_product_definition parameter of the **relation** argument.

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

5.5.4 acyclic_product_relationship

The **acyclic_product_relationship** function determines whether the graph of instances of the entity data type **product** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **product_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **product_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_product_relationship
  (relation          : product_relationship;
   relatives         : SET [1:?] OF product;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x          : SET OF product_relationship;
END_LOCAL;

IF relation.relatng_product IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (prod <* bag_to_set
            (USEDIN (relation.relatng_product,

```

```

        'PRODUCT_DEFINITION_SCHEMA.' +
        'PRODUCT_RELATIONSHIP.' +
        'RELATED_PRODUCT')) |
        specific_relation IN TYPEOF (prod));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_product_relationship
    (x[i],
     relatives + relation.relativing_product,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_product_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **product** that the function is searching for in the **relating_product** parameter of the **relation** argument.

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

5.5.5 get_product_definitions

The **get_product_definitions** function returns for any instance of **product** the set of instances of **product_definition** that refer to the product through an instance of **product_definition_formation**.

NOTE This function is not used in this schema.

EXPRESS specification

```

*)
FUNCTION get_product_definitions
  ( c_def_instance : product ) : SET OF product_definition;

  LOCAL
    pd_set : SET OF product_definition_formation := [];
    pdr_set : SET OF product_definition := [];
  END_LOCAL;

  pd_set := bag_to_set (USEDIN (c_def_instance,
    'PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION_FORMATION.OF_PRODUCT' ));

  IF (SIZEOF (pd_set) < 1 ) THEN RETURN (pdr_set);
  END_IF;
  (*
    Return empty SET, if no property_definition_formation is found.
  *)
  REPEAT i:= 1 to HIINDEX (pd_set);
    pdr_set := pdr_set + bag_to_set (USEDIN (pd_set[i],
      'PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION_FORMATION' ));
  END_REPEAT;
  RETURN (pdr_set);
END_FUNCTION;
(*

```

Argument definitions:

c_def_instance: (input) the candidate **product** to be checked.

EXPRESS specification

```
*)
END_SCHEMA; -- product_definition_schema
(*
```

6 Product property definition

The following EXPRESS declaration begins the **product_property_definition_schema** and identifies the necessary external references.

EXPRESS specification

```
*)
SCHEMA product_property_definition_schema;

REFERENCE FROM process_property_schema -- ISO 10303-49
  (action_property,
   resource_property);

REFERENCE FROM product_definition_schema -- ISO 10303-41
  (product_definition,
   product_definition_relationship);

REFERENCE FROM support_resource_schema -- ISO 10303-41
  (bag_to_set,
   identifier,
   label,
   text);
REFERENCE FROM basic_attribute_schema -- ISO 10303-41
  (get_id_value,
   id_attribute);
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

process_property_schema	ISO 10303-49
product_definition_schema	clause 5 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema is one of the generic product description resource schemas. The relationships between the generic product description resource schemas are given in annex E.1.

6.1 Introduction

The subject of the **product_property_definition_schema** is the characteristics of an object.

6.2 Fundamental concepts and assumptions

This schema provides resources for the identification of:

- types of properties to be identified and described;
- properties assigned to a **characterized_object**, a **product_definition**, a **product_definition_relationship** or a **shape_definition**.

The meaning of each assigned property may be defined either by its association with an instance of the entity characterizing a type of property or by its name attribute.

EXAMPLE 1 The definition of the surface finish of a shape is independent of the way in which the shape or the surface finish is represented.

NOTE In the previous example, both the defined surface finish and the shape could have multiple representations.

Each characteristic may be associated with a single **characterized_object** or **product_definition** or with one **product_definition** in the context of another. Each characteristic is independent of the number or types of representations of that characteristic. Each characteristic may be associated with the shape of a product, an element of the shape of a product, or the relationship between elements of the shape of a product.

This part of ISO 10303 distinguishes between the definition of a product and its possible usages. This part of ISO 10303 establishes the following assumptions:

- the definition of a given object is characterized by a set of unique properties.

EXAMPLE 2 A product cannot have two shapes simultaneously.

- any usage of the object is characterized by a set of unique properties.

EXAMPLE 3 A product, like glue, may have different shapes depending on its usage.

- a property characterizes either the definition or one of the usages of an object.

EXAMPLE 4 The appearance of chair x is a unique property of that chair. The colour designating that the chair is white is a single item in a representation for the appearance property of chair x. This colour is shareable among many representations for the properties of many different objects.

6.3 Product property definition type definitions

6.3.1 characterized_definition

The **characterized_definition** type allows for the designation of a **characterized_object**, a **characterized_product_definition** or a **shape_definition** to which properties may be assigned.

EXPRESS specification

```
*)
TYPE characterized_definition = SELECT
  (characterized_object,
   characterized_product_definition,
   shape_definition);
END_TYPE; -- characterized_definition
(*
```

6.3.2 characterized_product_definition

The **characterized_product_definition** type allows for the designation of a **product_definition** or a **product_definition_relationship** to which properties may be assigned.

Selection of a **product_definition_relationship** means that the property is applied to the related_
product_definition attribute in the context of its relating_product_definition attribute.

NOTE This enables properties of a given product that depend on the usages of this product to be described.

EXAMPLE The shape of a gasket depends upon whether or not it is an element of an assembly and, if it is an element of an assembly, the shape depends upon the assembly in which it participates.

EXPRESS specification

```
*)
TYPE characterized_product_definition = SELECT
  (product_definition,
   product_definition_relationship);
END_TYPE; -- characterized_product_definition
(*
```

6.3.3 derived_property_select

The **derived_property_select** type allows for the selection of a **property_definition**, an **action_property**, or a **resource_property**.

EXPRESS specification

```
*)
TYPE derived_property_select = SELECT
  (property_definition,
   action_property,
   resource_property);
END_TYPE; -- derived_property_select
(*
```

6.3.4 shape_definition

The **shape_definition** type allows for the designation of a **product_definition_shape**, a **shape_aspect**, or a **shape_aspect_relationship**.

References to a **shape_aspect_relationship** are references to the relationship itself and not to one of the associated **shape_aspects**.

EXPRESS specification

```

*)
TYPE shape_definition = SELECT
  (product_definition_shape,
   shape_aspect,
   shape_aspect_relationship);
END_TYPE; -- shape_definition
( *

```

6.4 Product property definition entity definitions**6.4.1 characterized_object**

A **characterized_object** is the identification of an item not deemed a **product** (see 5.4.1) that has associated property information.

NOTE 1 A **characterized_object** is characterized by the properties which refer to it.

NOTE 2 The properties of a **characterized_object** may be used as an environmental condition under which the properties of a product are measured.

EXAMPLE If a product has a set of properties that are measured within a room or an atmosphere, the room or atmosphere may be described with instances of the entity data type **characterized_object**.

EXPRESS specification

```

*)
ENTITY characterized_object;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- characterized_object
( *

```

Attribute definitions

name: the **label** by which the **characterized_object** is known.

description: the **text** that characterizes the **characterized_object**. The value of the attribute need not be specified.

6.4.2 characterized_object_relationship

A **characterized_object_relationship** relates two instances of the entity data type **characterized_object** and provides an identification and description of this relationship.

NOTE 1 The role of **characterized_object_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **characterized_object** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY characterized_object_relationship;
  name          : label;
  description   : OPTIONAL text;
  relating_object : characterized_object;
  related_object : characterized_object;
END_ENTITY; -- characterized_object_relationship
(*
```

Attribute definitions

name: the **label** by which the **characterized_object_relationship** is known.

description: the **text** that characterizes the **characterized_object_relationship**. The value of the attribute need not be specified.

relating_object: one of the instances of **characterized_object** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_object: the other instance of **characterized_object** that is a part of the relationship. If one element of the relationship is dependent upon the other, attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

6.4.3 general_property

A **general_property** identifies a type of property.

NOTE 1 An annotated Express schema can use this entity to identify a property selected in a library of properties.

NOTE 2 If needed, a **general_property** may be a member of a group.

EXAMPLE 1 "Kinematic viscosity", defined in ISO 31, is an example of a **general_property**.

EXPRESS specification

```
*)
ENTITY general_property;
  id          : identifier;
  name       : label;
  description : OPTIONAL text;
END_ENTITY; -- general_property
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **general_property**.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 4 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **general_property** is known.

EXAMPLE 2 'Kinematic viscosity' is an example of name.

description: the **text** that characterizes the **general_property**. The value of the attribute need not be specified.

6.4.4 general_property_association

A **general_property_association** is an association of an instance of one of the types listed in **derived_property_select** with a **general_property**. The **general_property** serves as the reference definition for the considered property.

EXAMPLE In order to characterize the kinematic viscosity of a fluid, an instance of **general_property_association** would relate an instance of **general_property** that defines the concept of kinematic viscosity and an instance of **property_definition** that defines the kinematic characteristics of an instance of **product_definition** defining the fluid.

EXPRESS specification

```
*)
ENTITY general_property_association;
  name          : label;
  description    : OPTIONAL text;
  base_definition : general_property;
  derived_definition : derived_property_select;
WHERE
  WR1: SIZEOF( USEDIN( derived_definition,
    'PRODUCT_PROPERTY_DEFINITION_SCHEMA.' +
    'GENERAL_PROPERTY_ASSOCIATION.' +
    'DERIVED_DEFINITION')) = 1;
  WR2: derived_definition.name = base_definition.name;
END_ENTITY; -- general_property_association
(*
```

Attribute definitions

name: the **label** by which the **general_property_association** is known.

description: the **text** that characterizes the **general_property_association**. The value of the attribute need not be specified.

base_definition: the **general_property** that provides the definition of the considered characteristic.

derived_definition: the **derived_property_select**, for which the identification of the considered characteristic is provided by a **general_property**.

Formal propositions:

WR1: The **derived_definition** shall be referred to by at most one **general_property_association**.

WR2: The attribute name of the `derived_definition` shall be the same as the attribute name of the `base_definition`.

6.4.5 `general_property_relationship`

A **`general_property_relationship`** relates two instances of the entity data type **`general_property`** and provides an identification and description of this relationship.

NOTE 1 This entity, together with the **`general_property`** entity, is based on the relationship template that is described in annex E.3

EXPRESS specification

```
* )
ENTITY general_property_relationship;
  name           : label;
  description    : OPTIONAL text;
  relating_property : general_property;
  related_property  : general_property;
END_ENTITY; -- general_property_relationship
(*
```

Attribute definitions

name: the **label** by which the **`general_property_relationship`** is known.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **`general_property_relationship`**. The value of the attribute need not be specified.

relating_property: one of the instance of **`general_property`** that is a part of the relationship.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_property: the other instance of **`general_property`** that is a part of the relationship. If one element of the relationship is dependent upon the other this attribute shall be the dependent one.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

6.4.6 `product_definition_shape`

A **`product_definition_shape`** is a type of **`property_definition`**. It identifies the shape of an object. A **`product_definition_shape`** identifies the shape of a **`characterized_object`** or of one of the types reachable as **`characterized_product_definition`**.

NOTE 1 A **`product_definition_shape`** need not be associated with any geometric representation.

NOTE 2 Early in the design of a product there may not be a specific idea about the shape of the product but there may be certain characteristics of the shape that are to be represented. Those product shape characteristics can be attached to the product shape using this entity.

EXAMPLE A geometric representation of shape is not needed to assert facts such as, "a shape must fit within a 5 centimetre cube".

EXPRESS specification

```

*)
ENTITY product_definition_shape
  SUBTYPE OF (property_definition);
UNIQUE
  UR1: SELF\property_definition.definition;
WHERE
  WR1: SIZEOF
    (['PRODUCT_PROPERTY_DEFINITION_SCHEMA.CHARACTERIZED_PRODUCT_DEFINITION',
    'PRODUCT_PROPERTY_DEFINITION_SCHEMA.CHARACTERIZED_OBJECT'] *TYPEOF
    (SELF\property_definition.definition)) > 0;
END_ENTITY; -- product_definition_shape
(*)

```

Formal propositions:

UR1: There shall not be two instances of **product_definition_shape** referring to the same item.

WR1: The definition attribute shall not refer to a **shape_definition**.

6.4.7 property_definition

A **property_definition** is a property that characterizes a single object.

NOTE 1 The type of the characterized item is one of the entity types that may be selected directly or indirectly with **characterized_definition**.

NOTE 2 The role of **property_definition** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXPRESS specification

```

*)
ENTITY property_definition;
  name      : label;
  description : OPTIONAL text;
  definition : characterized_definition;
DERIVE
  id          : identifier := get_id_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- property_definition
(*)

```

Attribute definitions

name: the **label** by which the **property_definition** is known.

description: the **text** that characterizes the **property_definition**. The value of the attribute need not be specified.

definition: the item whose property is identified.

id: the **identifier** that distinguishes the **property_definition**. The value of this attribute need not be specified.

NOTE 3 This attribute is an upwardly compatible addition to **property_definition** as specified in ISO 10303-41:1994.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 5 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

Formal propositions:

WR1: Each **product_definition** shall be the identified_item in at most one **id_attribute**.

NOTE 6 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 7 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

6.4.8 shape_aspect

A **shape_aspect** is an identified element of the shape of an object.

EXAMPLE 1 Consider the **product_definition_shape** of a bolt. One might distinguish, as an element of this shape, the concept of the threaded portion of its shank. This portion of the shape could be specified using a **shape_aspect** entity so that other properties, such as surface finish, may be associated with it.

EXPRESS specification

```

*)
ENTITY shape_aspect;
  name           : label;
  description    : OPTIONAL text;
  of_shape      : product_definition_shape;
  product_definitional : LOGICAL;
DERIVE
  id             : identifier := get_id_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- shape_aspect
(*

```

Attribute definitions

name: the **label** by which the **shape_aspect** is known.

description: the **text** that characterizes the **shape_aspect**. The value of the attribute need not be specified.

of_shape: the **product_definition_shape** of which this entity is an aspect.

EXAMPLE 2 If the identified aspect is the threaded portion of a bolt's shank, this attribute would be the **product_definition_shape** of the bolt.

product_definitional: an indication that the **shape_aspect** is on the physical boundary of the **product_definition_shape**. If the value of this attribute is TRUE, the **shape_aspect** being identified is on such a boundary. If the value is FALSE, the **shape_aspect** being identified is not on such a boundary. If the value is UNKNOWN, it is not known whether or not the **shape_aspect** being identified is on such a boundary.

EXAMPLE 3 If the identified **shape_aspect** is the threaded portion of a bolt's shank, the value of this attribute would be TRUE. If it is the centre-line, the value would be FALSE.

id: the **identifier** that distinguishes the **shape_aspect**. The value of this attribute need not be specified.

NOTE 1 This attribute is an upwardly compatible addition to **shape_aspect** as specified in ISO 10303-41:1994.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 3 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

Formal propositions:

WR1: Each **shape_aspect** shall be the identified **item** in at most one **id_attribute**.

NOTE 4 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 5 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

6.4.9 **shape_aspect_relationship**

A **shape_aspect_relationship** relates two instances of the entity data type **shape_aspect** and provides an identification and description of their relationship

NOTE 1 The role of **shape_aspect_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 If one **shape_aspect** is part of another, this entity could be used to associate the two **shape_aspects**.

NOTE 3 Each **shape_aspect** entity may have different properties.

NOTE 4 No actual physical relationship is established between related instances of the entity data type **shape_aspect**.

EXAMPLE 1 A **shape_aspect_relationship** might relate two instances of the entity data type **shape_aspect** whose representations are the equivalent surfaces of a mould and a moulded product. The shape of the mould is not spatially related to the moulded product.

NOTE 5 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS

schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 6 This entity, together with the **shape_aspect** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```

*)
ENTITY shape_aspect_relationship;
  name           : label;
  description    : OPTIONAL text;
  relating_shape_aspect : shape_aspect;
  related_shape_aspect  : shape_aspect;
DERIVE
  id             : identifier := get_id_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- shape_aspect_relationship
(*

```

Attribute definitions

name: the **label** by which the **shape_aspect_relationship** is known.

description: the **text** that characterizes the **shape_aspect_relationship**. The value of the attribute need not be specified.

relating_shape_aspect: one of the instances of **shape_aspect** that is a part of the relationship.

NOTE 7 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 2 A **shape_aspect** that is a pocket with five faces would play the role of **relating_shape_aspect** in five instances of the entity data type **shape_aspect_relationship**: one per face.

related_shape_aspect: the other instance of **shape_aspect** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 8 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 3 In the previous example each of the five instances of the entity data type **shape_aspect_relationship** would have a different **shape_aspect** entity in the **related_shape_aspect** field. There would be one for each side and one for the bottom of the pocket.

id: the **identifier** that distinguishes the **shape_aspect_relationship**. The value of this attribute need not be specified.

NOTE 9 This attribute is an upwardly compatible addition to **shape_aspect_relationship** as specified in ISO 10303-41:1994.

NOTE 10 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 11 The context in which id is used as a discriminating characteristic can be identified in an annotated

Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

Formal propositions:

WR1: Each **shape_aspect_relationship** shall be the identified_item in at most one **id_attribute**.

NOTE 12 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 13 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

6.5 Product property definition function definitions

6.5.1 acyclic_characterized_object_relationship

The **acyclic_characterized_object_relationship** function determines whether the graph of instances of the entity data type **characterized_object** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **characterized_object_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **characterized_object_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_characterized_object_relationship
  (relation      : characterized_object_relationship;
   relatives     : SET [1:?] OF characterized_object;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x      : SET OF characterized_object_relationship;
END_LOCAL;

IF relation.relatering_object IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (ca <* bag_to_set
            (USEDIN (relation.relatering_object,
                    'PRODUCT_PROPERTY_DEFINITION_SCHEMA.' +
                    'CHARACTERIZED_OBJECT_RELATIONSHIP.' +
                    'RELATED_OBJECT')) |
            specific_relation IN TYPEOF (ca));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_characterized_object_relationship
    (x[i],
     relatives + relation.relatering_object,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);

```

```
END_FUNCTION; -- acyclic_characterized_object_relationship
(*
```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **characterized_object** that the function is searching for in the **relating_characterized_object** parameter of the relation argument.

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

6.5.2 acyclic_general_property_relationship

The **acyclic_general_property_relationship** function determines whether the graph of instances of the entity data type **general_property** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **general_property_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **general_property_relationship** entity include rules that use this function.

EXPRESS specification

```
*)
FUNCTION acyclic_general_property_relationship
  (relation          : general_property_relationship;
   relatives        : SET [1:?] OF general_property;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x          : SET OF general_property_relationship;
END_LOCAL;

IF relation.relying_property IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (genp <* bag_to_set
  (USEDIN (relation.relying_property,
  'PRODUCT_PROPERTY_DEFINITION_SCHEMA.' +
  'GENERAL_PROPERTY_RELATIONSHIP.' +
  'RELATED_PROPERTY')) |
  specific_relation IN TYPEOF (genp));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_general_property_relationship
    (x[i],
     relatives + relation.relying_property,
     specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
RETURN (TRUE);
END_FUNCTION; -- acyclic_general_property_relationship
```

(*

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **general_property** that the function is searching for in the **relating_general_property** parameter of the relation argument.

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

6.5.3 acyclic_shape_aspect_relationship

The **acyclic_shape_aspect_relationship** function determines whether the graph of instances of the entity data type **shape_aspect** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **shape_aspect_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **shape_aspect_relationship** entity include rules that use this function.

EXPRESS specification

*)

```

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

LOCAL
  x          : SET OF shape_aspect_relationship;
END_LOCAL;

IF relation.relying_shape_aspect IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (sa <* bag_to_set
  (USEDIN (relation.relying_shape_aspect,
    'PRODUCT_PROPERTY_DEFINITION_SCHEMA.' +
    'SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT')) |
  specific_relation IN TYPEOF (sa));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_shape_aspect_relationship
    (x[i],
     relatives + relation.relying_shape_aspect,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_shape_aspect_relationship
(*)

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **shape_aspect** that the function is searching for in the relating_shape_aspect parameter of the relation argument.

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

6.5.4 get_shape_aspects

The **get_shape_aspects** function returns for any instance of **characterized_definition** the set of instances of **shape_aspect** that refer to the instance of **characterized_definition** through an instance of **product_definition_shape**.

NOTE This function is not used in this schema.

EXPRESS specification

```

*)
FUNCTION get_shape_aspects
  ( c_def_instance : characterized_definition ) : SET OF shape_aspect;

  LOCAL
    pd_set : SET OF product_definition_shape := [];
    pdr_set : SET OF shape_aspect := [];
  END_LOCAL;

  pd_set := bag_to_set (QUERY(pd <* USEDIN (c_def_instance,
    'PRODUCT_PROPERTY_DEFINITION_SCHEMA.PROPERTY_DEFINITION.DEFINITION' )
    | 'PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
    TYPEOF(pd)));

  IF (SIZEOF (pd_set) < 1) THEN RETURN (pdr_set);
  END_IF;
  (*
    Return empty SET, if no property_definition is found.
  *)
  REPEAT i:= 1 to HINDEX (pd_set);
    pdr_set := pdr_set + bag_to_set (USEDIN (pd_set[i],
    'PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_ASPECT.OF_SHAPE' ));
  END_REPEAT;
  RETURN (pdr_set);
END_FUNCTION;
  (*

```

Argument definitions:

c_def_instance: (input) the candidate **characterized_definition** to be checked.

EXPRESS specification

```

*)
END_SCHEMA; -- product_property_definition_schema
  (*

```

7 Product property representation

The following EXPRESS declaration begins the **product_property_representation_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA product_property_representation_schema;

REFERENCE FROM basic_attribute_schema                -- ISO 10303-41
  (description_attribute,
   get_description_value,
   get_name_value,
   name_attribute);

REFERENCE FROM material_property_definition_schema    -- ISO 10303-45
  (property_definition_relationship);

REFERENCE FROM product_definition_schema             -- ISO 10303-41
  (product_definition,
   product_definition_relationship);

REFERENCE FROM product_property_definition_schema    -- ISO 10303-41
  (characterized_definition,
   general_property,
   product_definition_shape,
   property_definition,
   shape_aspect,
   shape_aspect_relationship);

REFERENCE FROM representation_schema                 -- ISO 10303-43
  (representation,
   representation_item,
   representation_relationship,
   using_representations);

REFERENCE FROM support_resource_schema               -- ISO 10303-41
  (bag_to_set,
   label,
   text);

(*

```

NOTE 1 The schemas referenced above are specified in the following parts of ISO10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
material_property_definition_schema	ISO 10303-45
product_definition_schema	clause 5 of this part of ISO 10303
product_property_definition_schema	clause 6 of this part of ISO 10303
representation_schema	ISO 10303-43
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema is one of the generic product description resource schemas. The relationships between the generic product description resource schemas are given in annex E.1.

7.1 Introduction

The subject of the **product_property_representation_schema** is the representation of properties.

The requirements addressed by this schema are:

- the ability to associate a representation with a **property_definition** that it represents;
- the ability to identify a **representation** as representing a shape;

the ability to identify relationships between the representations of shapes and to state that the identified relationships are the representations of zero, one or more instances of the entity data type **product_definition**.

7.2 Fundamental concepts and assumptions

The following assumptions apply to the **product_property_representation_schema**:

- this International Standard will include various types of representations ranging from simple collections of geometry to more elaborate collections of representations of different kinds of properties;
- it is possible to have more than one representation of a single property;
- a single representation may be used to represent a property of zero, one, or many instances of the entity data type **product_definition**;
- any property of a **product_definition** is an identifiable concept independent of how, or even if, it is represented.

NOTE Representation schemas are defined in other parts of ISO 10303. This schema provides a structure within which these resources can be brought together to represent the properties of a product.

7.3 Product property representation type definition: represented_definition

The **represented_definition** type allows for the designation of a **general_property**, a **property_definition**, a **property_definition_relationship**, a **shape_aspect**, or a **shape_aspect_relationship**.

EXPRESS specification

*)

```
TYPE represented_definition = SELECT
  (general_property,
   property_definition,
   property_definition_relationship,
   shape_aspect,
   shape_aspect_relationship);
```

```
END_TYPE; -- represented_definition
(*
```

7.4 Product property representation entity definitions

7.4.1 context_dependent_shape_representation

A **context_dependent_shape_representation** is the association of a **shape_representation_relationship** with a **product_definition_shape**. The **product_definition_shape** identifies the shape of a **product_definition** as it plays the role of the **related_product_definition** in a **product_definition_relationship**.

NOTE 1 The role of **context_dependent_shape_representation** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXAMPLE 1 A gasket's shape depends upon whether the gasket is participating in an assembly relationship and, if it is a part of an assembly, the product or products with which it is assembled.

EXAMPLE 2 The position of a bolt's shape depends upon the way it is positioned in a given assembly.

EXPRESS specification

```
*)
ENTITY context_dependent_shape_representation;
  representation_relation      : shape_representation_relationship;
  represented_product_relation : product_definition_shape;
DERIVE
  description      : text := get_description_value (SELF);
  name             : label := get_name_value (SELF);
WHERE
  WR1 : 'PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION_RELATIONSHIP'
        IN TYPEOF (SELF.represented_product_relation.definition);
  WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                        'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  WR3 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                        'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- context_dependent_shape_representation
(*
```

Attribute definitions

representation_relation: a **shape_representation_relationship** that is associated with the **product_definition_shape**.

represented_product_relation: a **product_definition_shape** that identifies the shape of the related **product_definition** in the context of a **product_definition_relationship**.

description: the **text** that characterizes the **context_dependent_shape_representation**. The value of the attribute need not be specified.

NOTE 2 This attribute is an upwardly compatible addition to **context_dependent_shape_representation** as specified in ISO 10303-41:1994.

name: the **label** by which the **context_dependent_shape_representation** is known.

NOTE 3 This attribute is an upwardly compatible addition to **context_dependent_shape_representation** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: The **represented_product_relation** shall identify the shape of a **product_definition_relationship**.

WR2: Each **context_dependent_shape_representation** shall be the **described_item** in at most one **description_attribute**.

NOTE 4 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

WR3: Each **context_dependent_shape_representation** shall be the **named_item** in at most one **name_attribute**.

NOTE 5 The **name_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 6 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

7.4.2 item_identified_representation_usage

An **item_identified_representation_usage** identifies a **representation_item** within a **representation** as being the element that describes a particular component or part of the property that is described by the **representation**.

EXAMPLE In an application protocol, an instance of **representation** describes the shape of a product. One element of the **representation** - a curve - represents the boundary of a hole in the product. This entity data type may be used to state that the curve describes the hole, in the context of the entire **representation** describing the shape of the product.

EXPRESS specification

```
* )
ENTITY item_identified_representation_usage;
    name                : label;
    description         : OPTIONAL text;
    definition          : represented_definition;
    used_representation : representation;
    identified_item     : representation_item;
WHERE
    WR1: SELF.used_representation IN
        using_representations(SELF.identified_item);
END_ENTITY; -- item_identified_representation_usage
(*
```

Attribute definitions

name: the **label** by which the **item_identified_representation_usage** is known.

description: the **text** that characterizes the **item_identified_representation_usage**. The value of the attribute need not be specified.

definition: the identification of the **general_property**, **property_definition**, **property_definition_**

relationship, **shape_aspect**, or **shape_aspect_relationship** that is represented.

used_representation: the **representation** that describes the property or aspect of which the definition is a part or component, and that contains the **identified_item** (directly or indirectly) as one of its items.

identified_item: the **representation_item** that describes the identified property or aspect of the representation.

Formal propositions:

WR1: The **representation_item** that is identified shall belong to the set of items of the **representation** referred to by the attribute **used_representation**, or shall participate in the definition of one of the items.

7.4.3 property_definition_representation

A **property_definition_representation** is an association between a **property_definition** and a representation of the property.

EXPRESS specification

```

*)
ENTITY property_definition_representation;
  definition          : represented_definition;
  used_representation : representation;
DERIVE
  description        : text := get_description_value (SELF);
  name               : label := get_name_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- property_definition_representation
(*)

```

Attribute definitions

definition: the identification of the **general_property**, **property_definition**, **property_definition_relationship**, **shape_aspect**, or **shape_aspect_relationship** that is represented.

used_representation: the **representation** of the property or aspect of the representation.

description: the **text** that characterizes the **property_definition_representation**. The value of the attribute need not be specified.

NOTE 1 This attribute is an upwardly compatible addition to **property_definition_representation** as specified in ISO 10303-41:1994.

name: the **label** by which the **property_definition_representation** is known.

NOTE 2 This attribute is an upwardly compatible addition to **property_definition_representation** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **property_definition_representation** shall be the **described_item** in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

WR2: Each **property_definition_representation** shall be the **named_item** in at most one **name_attribute**.

NOTE 4 The **name_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 5 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

7.4.4 shape_definition_representation

A **shape_definition_representation** is the association of a **shape_representation** with a **product_definition_shape** or with a **property_definition** characterizing a **shape_definition**.

EXPRESS specification

```
*)
ENTITY shape_definition_representation
  SUBTYPE OF (property_definition_representation);
WHERE
  WR1: ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF (SELF.definition))
    OR
    ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_DEFINITION' IN
        TYPEOF (SELF.definition.definition));
  WR2: 'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_REPRESENTATION' IN
        TYPEOF (SELF.used_representation);
END_ENTITY; -- shape_definition_representation
(*)
```

Formal propositions:

WR1: The inherited definition attribute shall be a **product_definition_shape** or a **property_definition** whose definition attribute is a **shape_definition**.

WR2: The inherited **used_representation** attribute shall be a **shape_representation**.

7.4.5 shape_representation

A **shape_representation** is a type of **representation** that represents a shape.

EXPRESS specification

```
*)
ENTITY shape_representation
  SUBTYPE OF (representation);
END_ENTITY; -- shape_representation
(*)
```

7.4.6 shape_representation_relationship

A **shape_representation_relationship** is a type of **representation_relationship** in which at least one of the instances of the entity data type **representation** is a **shape_representation**.

EXAMPLE The **representation** of the shape of a bolt may be related to the representation of a position if the bolt is a part of an assembly.

EXPRESS specification

```
*)
ENTITY shape_representation_relationship
  SUBTYPE OF (representation_relationship);
WHERE
  WR1: 'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_REPRESENTATION' IN
    (TYPEOF(SELFF\representation_relationship.rep_1) +
     TYPEOF(SELFF\representation_relationship.rep_2));
END_ENTITY; -- shape_representation_relationship
(*
```

Formal propositions:

WR1: At least one of the two representations in the **shape_representation_relationship** shall be a **shape_representation**.

7.5 Product property representation function definitions

7.5.1 relatives_of_product_definition

The **relatives_of_product_definitions** function finds all of the instances of the **product_definition** entity data type that are related to one or more elements of the **definition_set** argument. Only those relationships that are established by the subtype of the **product_definition_relationship** entity given in the **relation_subtype** argument are considered by this function.

EXPRESS specification

```
*)
FUNCTION relatives_of_product_definitions
  (definition_set : SET OF product_definition;
   relation_subtype : STRING) : SET OF product_definition;

FUNCTION local_relatives_of_product_definitions
  (definition_set : SET OF product_definition;
   total_definitions : SET OF product_definition;
   relation_subtype : STRING) : SET OF product_definition;

LOCAL
  local_def : SET OF product_definition := [];
  local_pdr : SET OF product_definition_relationship := [];
  local_total : SET OF product_definition := [];
END_LOCAL;

REPEAT i := 1 TO HIINDEX(definition_set);
  local_pdr := local_pdr +
    bag_to_set(USEDIN
      (definition_set[i],
       relation_subtype + '.RELATING_PRODUCT_DEFINITION'));
END_REPEAT;
```

```

REPEAT i := 1 TO HIINDEX(local_pdr);
  local_def := local_def + local_pdr[i].related_product_definition;
END_REPEAT;
IF (SIZEOF(local_def) - SIZEOF(total_definitions)) = 0 THEN
  RETURN (local_def);
ELSE
  local_total := total_definitions + local_def;
  RETURN(local_def +
    (local_relatives_of_product_definitions
    (local_def - total_definitions, local_total, relation_subtype)));
END_IF;
END_FUNCTION; -- local_relatives_of_product_definitions

RETURN (local_relatives_of_product_definitions
  (definition_set, definition_set, relation_subtype));
END_FUNCTION; -- relatives_of_product_definitions
(*)

```

Argument definitions:

definition_set: (input) the set of instances of the entity data type **product_definition** upon which operations are performed.

relation_subtype: the fully qualified name of an entity that is a subtype of the **product_definition_relationship** entity.

7.5.2 relatives_of_shape_representations

The **relatives_of_shape_representations** function finds all of the instances of the **shape_representation** entity data type that are related to one or more elements of the **shape_rep_set** argument.

NOTE This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **identification_assignment_relationship** entity include rules that use this function.

EXPRESS specification

```

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

FUNCTION local_relatives_of_shape_representations
  (shape_representation_set : SET OF shape_representation;
  total_reps      : SET OF shape_representation) : SET OF
  shape_representation;

LOCAL
  local_shape_rep : SET OF shape_representation := [];
  local_srr       : SET OF shape_representation_relationship := [];
  local_total     : SET OF shape_representation := [];
END_LOCAL;

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

```

```

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

RETURN (local_relatives_of_shape_representations
  (shape_representation_set, shape_representation_set));
END_FUNCTION; -- relatives_of_shape_representations
(*

```

Argument definitions:

shape_representation_set: (input) the set of instances of the entity data type **shape_representation** to be operated on.

7.5.3 get_property_definition_representations

The **get_property_definition_representations** function returns for any **characterized_definition** the set of **property_definition_representation** that refer to the **characterized_definition** through a **property_definition**.

EXPRESS specification

```

*)
FUNCTION get_property_definition_representations
  ( c_def_instance : characterized_definition ) :
  SET OF property_definition_representation;

  LOCAL
    pd_set : SET OF property_definition := [];
    pdr_set : SET OF property_definition_representation := [];
  END_LOCAL;

  pd_set := bag_to_set (USEDIN (c_def_instance,
    'PRODUCT_PROPERTY_DEFINITION_SCHEMA.PROPERTY_DEFINITION.DEFINITION' ));

  IF (SIZEOF (pd_set) < 1 ) THEN RETURN (pdr_set);
  END_IF;
  (*
    Return empty SET, if no property_definition is found.
  *)
  REPEAT i:= 1 to HIINDEX (pd_set);
    pdr_set := pdr_set + bag_to_set (USEDIN (pd_set[i],
    'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.
    PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ));
  END_REPEAT;
  RETURN (pdr_set);
END_FUNCTION;
(*

```

Argument definitions:

c_def_instance: (input) the candidate **characterized_definition** to be checked.

EXPRESS specification

```
*)
END_SCHEMA; -- product_property_representation_schema
(*
```

8 Management resources

The following EXPRESS declaration begins the **management_resources_schema** and identifies the necessary external references.

EXPRESS specification

```
*)
SCHEMA management_resources_schema;

REFERENCE FROM action_schema; -- ISO 10303-41

REFERENCE FROM application_context_schema
(library_context); -- ISO 10303-41

REFERENCE FROM approval_schema; -- ISO 10303-41

REFERENCE FROM basic_attribute_schema
(get_role,
object_role,
role_association); -- ISO 10303-41

REFERENCE FROM certification_schema; -- ISO 10303-41

REFERENCE FROM contract_schema; -- ISO 10303-41

REFERENCE FROM date_time_schema; -- ISO 10303-41

REFERENCE FROM document_schema; -- ISO 10303-41

REFERENCE FROM effectivity_schema; -- ISO 10303-41

REFERENCE FROM external_reference_schema
(external_source); -- ISO 10303-41

REFERENCE FROM group_schema; -- ISO 10303-41

REFERENCE FROM person_organization_schema; -- ISO 10303-41

REFERENCE FROM security_classification_schema; -- ISO 10303-41

REFERENCE FROM support_resource_schema; -- ISO 10303-41
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

action_schema	clause 10 of this part of ISO 10303
application_context_schema	clause 4 of this part of ISO 10303

approval_schema	clause 12 of this part of ISO 10303
basic_attribute_schema	clause 22 of this part of ISO 10303
certification_schema	clause 11 of this part of ISO 10303
contract_schema	clause 13 of this part of ISO 10303
date_time_schema	clause 16 of this part of ISO 10303
document_schema	clause 9 of this part of ISO 10303
effectivity_schema	clause 18 of this part of ISO 10303
external_reference_schema	clause 19 of this part of ISO 10303
group_schema	clause 17 of this part of ISO 10303
person_organization_schema	clause 15 of this part of ISO 10303
security_classification_schema	clause 14 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains generic management resources that conform to the template in annex E.2. The way in which generic management resources are used is also described in annex E.2.

8.1 Introduction

The subject of the **management_resources_schema** is the association of management data with other aspects of product data in specific application contexts.

8.2 Fundamental concepts and assumptions

The relationship between management type data and other aspects of product data is application-specific. Management data may be assigned to various kinds of product data. The assignment of management data can be defined in a generic manner and then specialized according to each considered application context.

NOTE In the following definitions, the expression "assigned product data" means "the product data that are referenced by subtypes of the generic abstract supertypes of this schema". Use of these generic constructs is defined in annex E.2.

8.3 Management resources type definition: attribute_type

The **attribute_type** data type allows for the designation of a label or a text that can be assigned to an attribute of an entity.

EXPRESS specification

```
*)
TYPE attribute_type = SELECT
  (label,
   text);
END_TYPE; -- attribute_type
(*
```

8.4 Management resources entity definitions

8.4.1 action_assignment

An **action_assignment** is an association of an **action** (see 10.4.1) with product data.

EXPRESS specification

```
*)
ENTITY action_assignment
  ABSTRACT SUPERTYPE;
  assigned_action : action;
DERIVE
  role           : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- action_assignment
(*
```

Attribute definitions

assigned_action: the instance of the **action** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **action_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **action_assignment** using a method that is upwardly compatible with ISO 10303-41:1994.

Formal propositions

WR1: Each **action_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 2 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.2 action_method_assignment

An **action_method_assignment** is an association of an **action_method** (see 10.4.3) with product data.

EXPRESS specification

```
*)
ENTITY action_method_assignment
  ABSTRACT SUPERTYPE;
  assigned_action_method : action_method;
```

```

    role                : action_method_role;
END_ENTITY; -- action_method_assignment
(*)

```

Attribute definitions

assigned_action_method: the instance of the **action_method** entity data type that is to be associated with the product data.

role: the **action_method_role** that specifies the purpose of the association of the **action_method_assignment** with product data.

NOTE The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

8.4.3 action_method_role

An **action_method_role** defines a role for an **action_method_assignment** and a description of that role.

EXAMPLE 'Process XYZ' is an **action_method**. An **action_method_assignment** assigns the **action_method** to the definition of a specific mechanical part. The **action_method_role** for the **action_method_assignment** is 'Process to mill mechanical part'.

EXPRESS specification

```

*)
ENTITY action_method_role;
    name          : label;
    description    : OPTIONAL text;
END_ENTITY; -- action_method_role
(*)

```

Attribute definitions

name: the **label** by which the **action_method_role** is known.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **action_method_role**. The value of the attribute need not be specified.

8.4.4 action_request_assignment

An **action_request_assignment** is an association of a **versioned_action_request** (see 10.4.14) with product data.

EXPRESS specification

```

*)
ENTITY action_request_assignment
    ABSTRACT SUPERTYPE;
    assigned_action_request : versioned_action_request;
DERIVE
    role                : object_role := get_role (SELF);
WHERE

```

```

WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                    'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- action_request_assignment
(*)

```

Attribute definitions

assigned_action_request: the instance of the **versioned_action_request** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **action_request_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **action_request_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

NOTE 2 The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

Formal propositions:

WR1: Each **action_request_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 3 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.5 approval_assignment

An **approval_assignment** is an association of an **approval** (see 12.3.1) with product data.

EXPRESS specification

```

*)
ENTITY approval_assignment
  ABSTRACT SUPERTYPE;
  assigned_approval : approval;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                    'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- approval_assignment
(*)

```

Attribute definitions

assigned_approval: the instance of the **approval** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **approval_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **approval_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

NOTE 2 The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

Formal propositions:

WR1: Each **approval_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 3 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.6 attribute_classification_assignment

An **attribute_classification_assignment** is an association of a class to an attribute of an existing entity in order to characterize it further. The meaning of the class of the existing attribute is determined by the role.

EXPRESS specification

```
*)
ENTITY attribute_classification_assignment
  ABSTRACT SUPERTYPE;
  assigned_class      : group;
  attribute_name      : label;
  role                : classification_role;
END_ENTITY; -- attribute_classification_assignment
(*
```

Attribute definitions

assigned_class: the **group** that serves as a classification.

attribute_name: the attribute for which the classification is defined. The **attribute_name** shall identify an attribute of the assigned entity.

role: the **classification_role** that specifies the purpose of the association of the **attribute_classification_assignment** with product data.

NOTE The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

8.4.7 attribute_value_assignment

An **attribute_value_assignment** is an association of an additional value to a string valued attribute of an existing entity. The relationship of the additional value to the existing attribute is determined by the role. The entity that has the attribute for which an additional value is specified is identified by specializations of **attribute_value_assignment**.

EXPRESS specification

```
*)
ENTITY attribute_value_assignment
  ABSTRACT SUPERTYPE;
  attribute_name      : label;
  attribute_value     : attribute_type;
  role                : attribute_value_role;
END_ENTITY; -- attribute_value_assignment
(*
```

Attribute definitions

attribute_name: the attribute for which the additional value is defined. The **attribute_name** shall identify an attribute of the assigned entity.

NOTE 1 The identified attribute may be defined in the entity or inherited from a supertype.

EXAMPLE 'name' and 'description' are examples for the attribute_name.

attribute_value: the **attribute_type** that contains, as **label**, the additional value for the attribute of the assigned entity specified by **attribute_name**.

role: the **attribute_value_role** that specifies the purpose of the association of the **attribute_value_assignment** with product data.

NOTE 2 The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

Informal propositions:

IP1: The type of **attribute_value_assignment.attribute_value** (label or text) shall be the same as the type of the attribute specified by **attribute_value_assignment.attribute_name**.

8.4.8 attribute_value_role

An **attribute_value_role** describes the purpose of an additional value assigned to an attribute of an entity.

EXPRESS specification

```
* )
ENTITY attribute_value_role;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- attribute_value_role
(*
```

Attribute definitions

name: the **label** by which the **attribute_value_role** is known.

EXAMPLE 'Additional marketing name', 'primary value', or 'translated' are examples for name.

description: the **text** that characterizes the **attribute_value_role**. The value of the attribute need not be specified.

8.4.9 certification_assignment

A **certification_assignment** is an association of a **certification** (see 11.3.1) with product data.

EXPRESS specification

```

*)
ENTITY certification_assignment
  ABSTRACT SUPERTYPE;
  assigned_certification : certification;
DERIVE
  role                    : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                      'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- certification_assignment
(*

```

Attribute definitions

assigned_certification: the instance of the **certification** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **certification_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **certification_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

Formal propositions:

WR1: Each **certification_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 2 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.10 classification_assignment

A **classification_assignment** is an association of a class to product data.

EXPRESS specification

```

*)
ENTITY classification_assignment
  ABSTRACT SUPERTYPE;
  assigned_class          : group;
  role                   : classification_role;
END_ENTITY; -- classification_assignment
(*

```

Attribute definitions

assigned_class: the **group** that serves as a class.

role: the **classification_role** that specifies the purpose of the association of the **classification_assignment** with product data.

NOTE The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

8.4.11 classification_role

A **classification_role** defines a role for a **classification_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY classification_role;
  name          : label;
  description   : OPTIONAL text;
END_ENTITY; -- classification_role
(*
```

Attribute definitions

name: the **label** by which the **classification_role** is known.

description: the **text** that characterizes the **classification_role**. The value of the attribute need not be specified.

8.4.12 contract_assignment

A **contract_assignment** is an association of a **contract** (see 13.3.1) with product data.

EXPRESS specification

```
*)
ENTITY contract_assignment
  ABSTRACT SUPERTYPE;
  assigned_contract : contract;
DERIVE
  role              : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- contract_assignment
(*
```

Attribute definitions

assigned_contract: the instance of the **contract** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **contract_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **contract_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

NOTE 2 The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

Formal propositions:

WR1: Each **contract_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 3 A template for constraining the population of the entity data types defined in the **basic_attribute_-schema** is described in Annex E.

8.4.13 **date_and_time_assignment**

A **date_and_time_assignment** is an association of a **date_and_time** (see 16.4.4) with product data.

EXPRESS specification

```
*)
ENTITY date_and_time_assignment
  ABSTRACT SUPERTYPE;
  assigned_date_and_time : date_and_time;
  role : date_time_role;
END_ENTITY; -- date_and_time_assignment
(*
```

Attribute definitions

assigned_date_and_time: the instance of the **date_and_time** entity data type that is to be associated with the product data.

role: the **date_time_role** that specifies the purpose of the association of the **date_and_time_assignment** with product data.

NOTE The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 'Completed on' is an example of role that may be used to characterize the fact that the assigned_date_and_time defines the date and time of that day when some action completed.

8.4.14 **date_assignment**

A **date_assignment** is an association of a **date** (see 16.4.3) with product data.

EXPRESS specification

```
*)
ENTITY date_assignment
  ABSTRACT SUPERTYPE;
  assigned_date : date;
  role : date_role;
END_ENTITY; -- date_assignment
(*
```

Attribute definitions

assigned_date: the **date** that is to be associated with the product data.

role: the **date_role** that specifies the purpose of the association of the **date_assignment** with product data.

NOTE The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 'Creation date' is an example of role that may be used to characterize the fact that the assigned product data has been created on the assigned_date.

8.4.15 document_reference

A **document_reference** is an association of a **document** (see 9.4.1) with product data.

EXPRESS specification

```

*)
ENTITY document_reference
  ABSTRACT SUPERTYPE;
  assigned_document : document;
  source            : label;
DERIVE
  role              : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- document_reference
(*

```

Attribute definitions

assigned_document: the instance of the **document** entity data type that is to be associated with product data.

source: the **label** stating the origination of the **assigned_document**.

EXAMPLE Engineering' and 'library' are examples of sources.

role: the **object_role** that specifies the purpose of the association of the **document_reference_role** with product.

NOTE 1 This attribute is an enhancement to the definition of **document_reference** using a method that is upwardly compatible with ISO 10303-41:1994

NOTE 2 The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

Formal propositions:

WR1: Each **document_reference** shall be the item_with_role in at most one **role_association**.

NOTE 3 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.16 document_usage_constraint_assignment

A **document_usage_constraint_assignment** is an association of a **document_usage_constraint** (see 9.4.6) with product data.

EXPRESS specification

```

*)
ENTITY document_usage_constraint_assignment
  ABSTRACT SUPERTYPE;
  assigned_document_usage : document_usage_constraint;
  role                    : document_usage_role;
END_ENTITY; -- document_usage_constraint_assignment
(*)

```

Attribute definitions

assigned_document_usage: the instance of the **document_usage_constraint** entity data type that is associated with the product data.

role: the **document_usage_role** that specifies the purpose of the association of the **document_usage_constraint_assignment** with product data.

NOTE 1 The values related to this attribute can be specified in an annotated EXPRESS schema that uses or specializes this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 2 Multiple constraints can be expressed by multiple instances of (a subtype of) **document_usage_constraint**. Such multiple instances can refer to the same instance of **document_usage_role** or to different instances.

8.4.17 document_usage_role

A **document_usage_role** defines a role for a **document_usage_constraint_assignment** and a description of that role.

EXPRESS specification

```

*)
ENTITY document_usage_role;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- document_usage_role
(*)

```

Attribute definitions

name: the **label** by which the **document_usage_role** is known.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE Examples of **document_usage_role.name** are 'applying process constraint' and 'applying material constraint.'

description: the **text** that characterizes the **document_usage_role**. The value of the attribute need not be specified.

8.4.18 effectivity_assignment

An **effectivity_assignment** is an association of an **effectivity** (see 18.3.2) with product data.

EXPRESS specification

```

*)
ENTITY effectivity_assignment
  ABSTRACT SUPERTYPE;
  assigned_effectivity : effectivity;
DERIVE
  role                  : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- effectivity_assignment
(*

```

Attribute definitions

assigned_effectivity: the instance of the **effectivity** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **effectivity_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **effectivity_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

Formal propositions:

WR1: Each **effectivity_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 2 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.19 effectivity_context_assignment

An **effectivity_context_assignment** identifies a context for the association of an **effectivity_assignment** with product data.

EXPRESS specification

```

*)
ENTITY effectivity_context_assignment
  ABSTRACT SUPERTYPE;
  assigned_effectivity_assignment : effectivity_assignment;
  role                           : effectivity_context_role;
END_ENTITY; -- effectivity_context_assignment
(*

```

Attribute definitions

assigned_effectivity_assignment: the instance of the **effectivity_assignment** entity data type that is to be associated with the product data.

role: the **effectivity_context_role** that specifies the purpose of the association of the **effectivity_context_assignment** with product data.

8.4.20 effectivity_context_role

An **effectivity_context_role** defines a role for an **effectivity_context_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY effectivity_context_role;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- effectivity_context_role
(*
```

Attribute definitions

name: the **label** by which the **effectivity_context_role** is known.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 'Manufacturing location' is an example of the name of an **effectivity_context_role** that may be used to characterize the fact that if effectivity is assigned to product data in the context of a particular plant, this plant plays the role of a manufacturing location.

description: the **text** that characterizes the **effectivity_context_role**. The value of the attribute need not be specified.

8.4.21 event_occurrence_assignment

An **event_occurrence_assignment** is an association of an **event_occurrence** with product data.

EXPRESS specification

```
*)
ENTITY event_occurrence_assignment
  ABSTRACT SUPERTYPE;
  assigned_event_occurrence : event_occurrence;
  role                      : event_occurrence_role;
END_ENTITY; -- event_occurrence_assignment
(*
```

Attribute definitions

assigned_event_occurrence: the instance of the **event_occurrence** entity data type that is to be associated with product data.

role: the **event_occurrence_role** that specifies the purpose of the associations of the **event_occurrence_assignment** with product data.

8.4.22 event_occurrence_context_assignment

An **event_occurrence_context_assignment** identifies a context for the association of an **event_occurrence_assignment** with product data.

EXAMPLE For the **event_occurrence** 'start of production', the **product** for which production starts is the context for that **event_occurrence**.

EXPRESS specification

```
*)
ENTITY event_occurrence_context_assignment
  ABSTRACT SUPERTYPE;
  assigned_event_occurrence_assignment : event_occurrence_assignment;
  role : event_occurrence_context_role;
END_ENTITY; -- event_occurrence_context_assignment
(*
```

Attribute definitions

assigned_event_occurrence_assignment: the instance of the **event_occurrence_assignment** entity data type that is to be associated with product data.

role: the **event_occurrence_context_role** that specifies the purpose of the association of the **event_occurrence_context_assignment** with product data.

8.4.23 external_identification_assignment

An **external_identification_assignment** is a type of **identification_assignment** that is defined in the context of an **external_source** (see 19.4.1).

The **source_id** shall be a valid identifier in the context of the source.

EXPRESS specification

```
*)
ENTITY external_identification_assignment
  ABSTRACT SUPERTYPE
  SUBTYPE OF (identification_assignment);
  source : external_source;
END_ENTITY; -- external_identification_assignment
(*
```

Attribute definitions

source: the **external_source** that specifies the context in which in the **source_id** is defined.

8.4.24 external_referent_assignment

An **external_referent_assignment** is an identification of product data referenced from outside sources.

EXPRESS specification

```

*)
ENTITY external_referent_assignment
  ABSTRACT SUPERTYPE;
  assigned_name : label;
DERIVE
  role          : object_role := get_role (SELF);
UNIQUE
  UR1 : assigned_name;
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- external_referent_assignment
(*)

```

Attribute definitions

assigned_name: the **label** by which the **external_referent_assignment** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

role: the **object_role** that specifies the purpose of the association of the **external_referent_assignment** with product data.

NOTE 2 This attribute is an enhancement to the definition of **external_referent_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

Formal propositions:

UR1: The **assigned_name** shall be unique.

WR1: Each **external_referent_assignment** shall be the **item_with_role** in at most one **role-association**.

NOTE 3 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.25 group_assignment

A **group_assignment** is an association of a **group** (see 17.3.1) with product data.

EXPRESS specification

```

*)
ENTITY group_assignment
  ABSTRACT SUPERTYPE;
  assigned_group : group;
DERIVE
  role          : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- group_assignment
(*)

```

Attribute definitions

assigned_group: the instance of the **group** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **group_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **group_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

Formal propositions:

WR1: Each **group_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 2 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.26 identification_assignment

An **identification_assignment** is an association of an identification with product data. The assignment is made within a specified role.

EXPRESS specification

```
* )
ENTITY identification_assignment
  ABSTRACT SUPERTYPE;
  assigned_id : identifier;
  role       : identification_role;
END_ENTITY; -- identification_assignment
(*
```

Attribute definitions

assigned_id: the identification associated with the product data.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

role: the **identification_role** that specifies the purpose of the association of the **identification_assignment** with product data.

EXAMPLE 'Alias identification' is an example of a role that may be used to characterize the fact that the **assigned_id** provides an alternate identification to some product data.

8.4.27 identification_assignment_relationship

An **identification_assignment_relationship** relates two instances of the entity data type **identification_assignment** and provides an identification and description of their relationship.

NOTE 1 The role of **identification_assignment_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **identification_assignment** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY identification_assignment_relationship;
  name                : label;
  description         : OPTIONAL text;
  relating_identification_assignment : identification_assignment;
  related_identification_assignment : identification_assignment;
END_ENTITY; -- identification_assignment_relationship
(*
```

Attribute definitions

name: the **label** by which the **identification_assignment_relationship** is known.

description: the **text** that characterizes the **identification_assignment_relationship**. The value of the attribute need not be specified.

relating_identification_assignment: one of the instances of **identification_assignment** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_identification_assignment: the other instance of **identification_assignment** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

8.4.28 **identification_role**

An **identification_role** defines a role for an **identification_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY identification_role;
  name          : label;
  description   : OPTIONAL text;
END_ENTITY; -- identification_role
(*
```

Attribute definitions

name: the **label** by which the **identification_role** is known.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **identification_role**. The value of the attribute need not be specified.

8.4.29 library_assignment

A **library_assignment** is a type of **external_referent_assignment** that assigns a **library_context** to an externally referenced document.

EXPRESS specification

```
*)
ENTITY library_assignment
  ABSTRACT SUPERTYPE
  SUBTYPE OF (external_referent_assignment);
  frame_of_reference : library_context;
UNIQUE
  URL: frame_of_reference;
END_ENTITY; -- library_assignment
(*
```

Attribute definitions

frame_of_reference: the **library_context** in which the **library_assignment** is defined.

Formal propositions:

URI: The **frame_of_reference** shall be unique.

8.4.30 name_assignment

A **name_assignment** is an identification of a name used to designate product data.

EXPRESS specification

```
*)
ENTITY name_assignment
  ABSTRACT SUPERTYPE;
  assigned_name : label;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- name_assignment
(*
```

Attribute definitions

assigned_name: the **label** by which the product data is known.

role: the **object_role** that specifies the purpose of the association of the **name_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **name_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

Formal propositions:

WR1: Each **name_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 2 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.31 organization_assignment

An **organization_assignment** is an association of an **organization** (see 15.4.2) with product data.

EXPRESS specification

```
*)
ENTITY organization_assignment
  ABSTRACT SUPERTYPE;
  assigned_organization : organization;
  role                  : organization_role;
END_ENTITY; -- organization_assignment
(*
```

Attribute definitions

assigned_organization: the instance of the **organization** entity data type that is to be associated with the product data.

role: the **organization_role** that specifies the purpose of the association of the **organization_assignment** with product data.

EXAMPLE 'Supplier' is an example of a role that may be used to characterize the fact that the assigned product data has been released by the assigned_organization.

8.4.32 organizational_project_assignment

An **organizational_project_assignment** is an association of an **organizational_project** with product data.

EXPRESS specification

```
*)
ENTITY organizational_project_assignment
  ABSTRACT SUPERTYPE;
  assigned_organizational_project : organizational_project;
  role                           : organizational_project_role;
END_ENTITY; -- organizational_project_assignment
(*
```

Attribute definitions

assigned_organizational_project: the instance of the **organizational_project** entity data type that is to be associated with product data.

role: the **organizational_project_role** that specifies the purpose of the association of the **organizational_project_assignment** with product data.

8.4.33 **organizational_project_role**

An **organizational_project_role** defines a role for an **organizational_project_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY organizational_project_role;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- organizational_project_role
(*
```

Attribute definitions

name: the **label** by which the **organizational_project_role** is known.

description: the **text** that characterizes the **organizational_project_role**.

8.4.34 **person_and_organization_assignment**

A **person_and_organization_assignment** is an association of a **person_and_organization** (see 15.4.9) with product data.

EXPRESS specification

```
*)
ENTITY person_and_organization_assignment
  ABSTRACT SUPERTYPE;
  assigned_person_and_organization : person_and_organization;
  role                             : person_and_organization_role;
END_ENTITY; -- person_and_organization_assignment
(*
```

Attribute definitions

assigned_person_and_organization: the instance of the **person_and_organization** entity data type that is to be associated with the product data.

role: the **person_and_organization_role** that specifies the purpose of the association of the **person_and_organization_assignment** with product data.

EXAMPLE 'Contact person' is an example of a role that may be used to characterize the fact that information about the assigned product data can be obtained with contacting the mentioned person in the specified organization.

8.4.35 **person_assignment**

A **person_assignment** is the association of a **person** (15.4.8) with product data.

EXPRESS specification

```
*)
ENTITY person_assignment
  ABSTRACT SUPERTYPE;
```

```

    assigned_person : person;
    role             : person_role;
END_ENTITY; -- person_assignment
(*)

```

Attribute definitions

assigned_person: the instance of the **person** entity data type that is to be associated with product data.

role: the **person_role** that specifies the purpose of the association of the **person_assignment** with product data.

NOTE The function of the **person** with respect to the assignment is usually a consequence of the functions of the **person** within his organization.

EXAMPLE 'Designer' and 'creator' are examples of roles that may be used to characterize the fact that the assigned product data has been designed or created by the **assigned_person**.

8.4.36 security_classification_assignment

A **security_classification_assignment** is an association of a **security_classification** (see 14.3.1) with product data.

EXPRESS specification

```

*)
ENTITY security_classification_assignment
  ABSTRACT SUPERTYPE;
  assigned_security_classification : security_classification;
DERIVE
  role : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                      'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- security_classification_assignment
(*)

```

Attribute definitions

assigned_security_classification: the instance of the **security_classification** entity data type that is to be associated with product data.

role: the **object_role** that specifies the purpose of the association of the **security_classification_assignment** with product data.

NOTE 1 This attribute is an enhancement to the definition of **security_classification_assignment** using a method that is upwardly compatible with ISO 10303-41:1994

Formal propositions:

WR1: Each **security_classification_assignment** shall be the **item_with_role** in at most one **role_association**.

NOTE 2 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

8.4.37 time_assignment

A **time_assignment** is an association of a **local_time** (see 16.4.11) with product data.

EXPRESS specification

```
*)
ENTITY time_assignment
  ABSTRACT SUPERTYPE;
  assigned_time : local_time;
  role          : time_role;
END_ENTITY; -- time_assignment
(*
```

Attribute definitions

assigned_time: the instance of the **local_time** entity data type that is to be associated with product data.

role: the **time_role** that specifies the purpose of the association of the **time_assignment** with product data.

EXAMPLE 'Start time' is an example of a role that may be used to characterize the fact that the assigned_time defines the time when some action was expected to start.

8.4.38 time_interval_assignment

A **time_interval_assignment** is an association of a **time_interval** with product data.

EXPRESS specification

```
*)
ENTITY time_interval_assignment
  ABSTRACT SUPERTYPE;
  assigned_time_interval : time_interval;
  role                  : time_interval_role;
END_ENTITY; -- time_interval_assignment
(*
```

Attribute definitions

assigned_time_interval: the instance of the **time_interval** entity data type that is to be associated with product data.

role: the **time_interval_role** that specifies the purpose of the association of the **time_interval_assignment** with product data.

8.5 Management resources function definition: acyclic_identification_assignment_relationship

The **acyclic_identification_assignment_relationship** function determines whether the graph of instances of the entity data type **identification_assignment** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **identification_assignment_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **identification_assignment_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_identification_assignment_relationship
  (relation          : identification_assignment_relationship;
   relatives         : SET [1:?] OF identification_assignment;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x          : SET OF identification_assignment_relationship;
END_LOCAL;

IF relation.relying_identification_assignment IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (ia <* bag_to_set
            (USEDIN (relation.relying_identification_assignment,
                    'MANAGEMENT_RESOURCES_SCHEMA.' +
                    'IDENTIFICATION_ASSIGNMENT_RELATIONSHIP.' +
                    'RELATED_IDENTIFICATION_ASSIGNMENT')) |
            specific_relation IN TYPEOF (ia));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_identification_assignment_relationship
    (x[i],
     relatives + relation.relying_identification_assignment,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_identification_assignment_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **identification_assignment** that the function is searching for in the **relying_identification_assignment** parameter of the relation argument.

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

EXPRESS specification

```

*)
END_SCHEMA; -- management_resources_schema
(*

```

9 Document

The following EXPRESS declaration begins the **document_schema** and identifies the necessary external references.

EXPRESS specification

```
* )
SCHEMA document_schema ;

REFERENCE FROM product_definition_schema          -- ISO 10303-41
( product ,
  product_definition ,
  product_definition_formation ) ;

REFERENCE FROM support_resource_schema           -- ISO 10303-41
( bag_to_set ,
  identifier ,
  label ,
  text ) ;
( *
```

NOTE 1 The schemas referenced above are specified in the following part of ISO 10303:

product_definition_schema	clause 5 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

9.1 Introduction

The subject of the **document_schema** is the citations of formal standards or documents that are outside the domain of this International Standard. These resource constructs are used to reference additional information that is relevant to the description of the product.

EXAMPLE International, national, and organizational standards, catalogues, and tables of engineering data are examples of formal standards or documents.

9.2 Fundamental concepts and assumptions

Product data can include citations of other information sources.

EXAMPLE Heat treatment processes documentation can be specified using the resource constructs that are defined in this schema.

9.3 Document type definition: product_or_formation_or_definition

The **product_or_formation_or_definition** type allows for the designation of a **product**, a **product_definition_formation**, or a **product_definition**.

EXPRESS specification

```
*)
TYPE product_or_formation_or_definition = SELECT
  (product,
   product_definition_formation,
   product_definition);
END_TYPE; -- product_or_formation_or_definition
(*
```

9.4 Document entity definitions

9.4.1 document

A **document** identifies a collection of information.

NOTE 1 The information collected need not be represented in a format conforming to any EXPRESS schema.

EXAMPLE JPEG and HTML files are examples of document.

NOTE 2 This part of ISO 10303 provides several resources to describe documents and to associate them with product data. Annex F.2 provides an explanation about how to describe documents if they are to be considered products.

EXPRESS specification

```
*)
ENTITY document;
  id                : identifier;
  name              : label;
  description       : OPTIONAL text;
  kind              : document_type;
INVERSE
  representation_types : SET[0:?] OF document_representation_type
                        FOR represented_document;
END_ENTITY; -- document
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **document**.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 4 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **document** is known.

NOTE 5 The name may include the source of the **document**, e.g. ISO.

description: the **text** that characterizes the **document**. The value of the attribute need not be specified.

kind: the **document_type** that specifies the sort of data that the document describes.

representation_types: the set of **document_representation_type** entities that specifies the way a document is represented.

9.4.2 document_product_association

A **document_product_association** is an association between a **document** and a **product**, between a **document** and a **product_definition_formation**, or between a **document** and a **product_definition**.

NOTE 1 The role of **document_product_association** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 The **document_product_association** may establish an equivalence between the **document** and the **product_definition_formation** or between the **document** and the **product** of the **document** and the **product_definition**. This equivalence will provide a document with the versioning structure found in the **product_definition_schema**.

EXPRESS specification

```
*)  
ENTITY document_product_association;  
  name          : label;  
  description    : OPTIONAL text;  
  relating_document : document;  
  related_product : product_or_formation_or_definition;  
END_ENTITY; -- document_product_association  
(*
```

Attribute definitions

name: the **label** by which the **document_product_association** is known.

description: the **text** that characterizes the **document_product_association**. The value of the attribute need not be specified.

relating_document: the instance of **document** that is part of the association.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_product: the instance of **product**, **product_definition_formation** or **product_definition** entity data type associated with an instance of **document**.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

9.4.3 document_relationship

A **document_relationship** relates two instances of the entity data type **document** and provides a description of their relationship.

NOTE 1 The role of **document_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **document** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY document_relationship;
  name          : label;
  description   : OPTIONAL text;
  relating_document : document;
  related_document : document;
END_ENTITY; -- document_relationship
(*
```

Attribute definitions

name: the **label** by which the **document_relationship** is known.

description: the **text** that characterizes the **document_relationship**. The value of the attribute need not be specified.

relating_document: one of the instances of **document** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_document: the other instance of **document** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

9.4.4 document_representation_type

A **document_representation_type** specifies the way a document is represented.

EXPRESS specification

```
*)
ENTITY document_representation_type;
  name          : label;
  represented_document : document;
END_ENTITY; -- document_representation_type
(*
```

Attribute definitions

name: the **label** by which the **document_representation_type** is known.

EXAMPLE 'Digital' and 'physical' are examples for the name.

represented_document: the **document** for which the kinds of representation are specified.

9.4.5 **document_type**

A **document_type** is the kind of data that a formal standard or document may provide information about.

NOTE The applicable values of **document_type** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXAMPLE 1 'Material', 'surface finish', and 'heat treatment process' are all pieces of data that can be described implicitly, by reference to other documents (such as standards documents), rather than explicitly every time they are used.

EXPRESS specification

```
* )
ENTITY document_type;
    product_data_type : label;
END_ENTITY; -- document_type
( *
```

Attribute definitions

product_data_type: the name of the kind of data contained in a document.

EXAMPLE 2 'Material description' is an example of **product_data_type** that characterizes the information provided by a standards document for material.

9.4.6 **document_usage_constraint**

A **document_usage_constraint** identifies a specific subject area or aspect from within a **document** and gives the relevant information or text that applies. The semantics of the reference can be found in the **document** itself.

NOTE 1 This entity could be used to identify the clause of a document and the content or portions of the content in that clause that are relevant.

EXPRESS specification

```
* )
ENTITY document_usage_constraint;
    source                : document;
    subject_element       : label;
    subject_element_value : text;
END_ENTITY; -- document_usage_constraint
( *
```

Attribute definitions

source: the **document** in which the **subject_element** originates

NOTE 2 The inverse of this relationship is used to define multiple elements of a **document**.

subject_element: the name of an element of the **source**.

subject_element_value: the **label** that conveys a specific value of the **subject_element** present in the source.

EXAMPLE For references to a 'surface finish' source the subject_element may be a 'Clause 2.1 surface imperfection' and its associated subject_element_value may be 'no visible imperfections', 'no more than two imperfections that are more than 0.06 inch diameter within any given square inch of surface area', or 'no imperfections under 10 times magnification'.

9.4.7 document_with_class

A **document_with_class** is a type of **document** for which a classification is specified.

EXAMPLE 1 A surface finish document may be identified by various classes, class A, class B, and class C. Each class of surface finish specifies various allowances for imperfections in the surface finish. Class A may require no visible imperfections, class B may require no more than two imperfections that are more than 0.06 inch diameter within any given square inch of surface area, and class C may require no imperfections under 10 times magnification.

EXPRESS specification

```
* )
ENTITY document_with_class
  SUBTYPE OF (document);
  class : identifier;
END_ENTITY; -- document_with_class
(*
```

Attribute definitions

class: the identification of the data classification to which the **document** belongs.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 2 The value of this attribute would be 'A', 'B' or 'C' in the previous example.

9.5 Document function definition: acyclic_document_relationship

The **acyclic_document_relationship** function determines whether the graph of instances of the entity data type **document** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **document_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **document_relationship** entity include rules that use this function.

EXPRESS specification

```

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

LOCAL
  x          : SET OF document_relationship;
END_LOCAL;

IF relation.relating_document IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (doc <* bag_to_set
            (USEDIN (relation.relating_document,
                    'DOCUMENT_SCHEMA.' +
                    'DOCUMENT_RELATIONSHIP.' +
                    'RELATED_DOCUMENT')) |
            specific_relation IN TYPEOF (doc));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_document_relationship
    (x[i],
     relatives + relation.relating_document,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_document_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **document** for which the function is searching in the `relating_document` parameter of the relation argument.

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

EXPRESS specification

```

*)
END_SCHEMA; -- document_schema
(*

```

10 Action

The following EXPRESS declaration begins the **action_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA action_schema;

REFERENCE FROM basic_attribute_schema           -- ISO 10303-41
  (description_attribute,
   get_description_value,
   get_id_value,
   get_name_value,
   id_attribute,
   name_attribute);

REFERENCE FROM support_resource_schema         -- ISO 10303-41
  (bag_to_set,
   identifier,
   label,
   text);
(*)

```

NOTE 1 The schemas referenced above are specified in the following part of ISO 10303:

basic_attribute_schema clause 22 of this part of ISO 10303

support_resource_schema clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

10.1 Introduction

The subject of the **action_schema** is the description of actions, the reasons for these actions, and the status of these actions.

EXAMPLE Reasons for action include evolving user requirements, manufacturing problems and difficulties that arise when a product is in use.

10.2 Fundamental concepts and assumptions

Action information can be attached to any aspect of product data.

10.3 Action type definition: supported_item

The **supported_item** allows for the designation of an **action_directive**, an **action**, or an **action_method**.

NOTE This specifies the use of an **action_resource**.

EXPRESS specification

```

*)
TYPE supported_item = SELECT
  (action_directive,
   action,
   action_method);
END_TYPE; -- supported_item
(*

```

10.4 Action entity definitions

10.4.1 action

An **action** is the identification of the occurrence of an activity and a description of its result.

An **action** identifies an activity that has taken place, is taking place, or is expected to take place in the future.

An **action** has a definition that is specified by an **action_method**.

NOTE 1 In particular application domains, terms such as task, process, activity, operation, and event may be synonyms for **action**.

EXAMPLE Change, distilling, design, a process to drill a hole, and a task such as training someone are examples of actions.

EXPRESS specification

```

*)
ENTITY action;
  name          : label;
  description   : OPTIONAL text;
  chosen_method : action_method;
DERIVE
  id            : identifier := get_id_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- action
(*

```

Attribute definitions

name: the **label** by which the **action** is known.

description: the **text** that characterizes the **action**. The value of the attribute need not be specified.

chosen_method: the **action_method** that specifies the procedure selected to carry out the **action** and a description of the results of that **action**.

id: the **identifier** that distinguishes the **action**.

NOTE 2 This attribute is an enhancement to the definition of **action** using a method that is upwardly compatible with ISO 10303-41:1994

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 4 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

Formal propositions:

WR1: Each **action** shall be the identified_item in at most one **id_attribute**.

NOTE 5 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 6 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

10.4.2 action_directive

An **action_directive** is an authoritative instrument that provides directions to achieve the specified results.

EXAMPLE ISO Directives Part 3 provides guidance for the development of standards documents within ISO.

EXPRESS specification

```
*)
ENTITY action_directive;
  name          : label;
  description   : OPTIONAL text;
  analysis      : text;
  comment       : text;
  requests      : SET[1:?] OF versioned_action_request;
END_ENTITY; -- action_directive
(*
```

Attribute definitions

name: the **label** by which the **action_directive** is known.

description: the **text** that characterizes the **action_directive**. The value of the attribute need not be specified.

analysis: an informal description of the results of the analysis that was carried out on the elements of the requests set. The request sets are the **versioned_action_requests** referenced in the requests attribute.

NOTE The reason that different requests are satisfied by the **action_directive** could be recorded in this attribute.

comment: an informal description of any other pertinent information.

requests: a set of **versioned_action_request** entity data types that defines the expected results.

10.4.3 action_method

An **action_method** is the definition of an activity. This definition includes the activity's objectives and effects.

NOTE This definition may be the basis for actions or the solution for action requests.

EXAMPLE For the **action** whose name attribute is 'serve dinner', the name attribute of related instance of **action_method** could be 'cook by recipe' or 'purchase takeout food'.

EXPRESS specification

```
*)
ENTITY action_method;
  name          : label;
  description    : OPTIONAL text;
  consequence    : text;
  purpose        : text;
END_ENTITY; -- action_method
(*
```

Attribute definitions

name: the **label** by which the **action_method** is known.

description: the **text** that characterizes the **action_method**. The value of the attribute need not be specified.

consequence: an informal description of the effects of the **action_method**.

purpose: an informal description of the objectives of the **action_method**.

10.4.4 action_method_relationship

An **action_method_relationship** relates two instances of the entity data type **action_method** and provides an identification and description of this relationship.

NOTE 1 The role of **action_method_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 This entity, together with the **action_method** entity, is based on the relationship template that is described in annex E.3.

NOTE 3 This entity may be used to define a procedural relationship among constituent activities.

EXPRESS specification

```
*)
ENTITY action_method_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_method : action_method;
  related_method  : action_method;
END_ENTITY; -- action_method_relationship
(*
```

Attribute definitions

name: the **label** by which the **action_method_relationship** is known.

description: the **text** that characterizes the **action_method_relationship**. The value of the attribute need not be specified.

relating_method: one of the instances of **action_method** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_method: the other instance of **action_method** that is a part of the relationship. If one element of the relationship is dependent upon the other this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

10.4.5 **action_relationship**

An **action_relationship** relates two instances of the entity data type **action** and provides an identification and description of this relationship.

NOTE 1 The role of **action_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXPRESS specification

```
* )
ENTITY action_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_action : action;
  related_action : action;
END_ENTITY; -- action_relationship
(*
```

Attribute definitions

name: the **label** by which the **action_relationship** is known.

description: the **text** that characterizes the **action_relationship**. The value of the attribute need not be specified.

relating_action: one of the instances of **action** that is a part of the relationship.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_action: the other instance of **action** that is a part of the relationship. If one element of the relationship is dependent upon the other this attribute shall be the dependent one.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

10.4.6 action_request_solution

An **action_request_solution** is an association between a **versioned_action_request** and an **action_method** that is a potential solution for the request.

EXPRESS specification

```

*)
ENTITY action_request_solution;
  method      : action_method;
  request     : versioned_action_request;
DERIVE
  description : text := get_description_value (SELF);
  name        : label := get_name_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- action_request_solution
(*

```

Attribute definitions

method: the **action_method** that is a potential solution.

request: the **versioned_action_request** for which a solution is specified.

description: the **text** that characterizes the **action_request_solution**.

NOTE 1 This attribute is an upwardly compatible addition to **action_request_solution** as specified in ISO 10303-41:1994.

name: the **label** by which the **action_request_solution** is known.

NOTE 2 This attribute is an upwardly compatible addition to **action_request_solution** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **action_request_solution** shall be the **described_item** in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

WR2: Each **action_request_solution** shall be the **named_item** in at most one **name_attribute**.

NOTE 4 The **name_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 5 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

10.4.7 action_request_status

An **action_request_status** is the association of a status with an **versioned_action_request**.

EXPRESS specification

```

*)
ENTITY action_request_status;
  status      : label;
  assigned_request : versioned_action_request;
END_ENTITY; -- action_request_status
(*

```

Attribute definitions

status: the **label** that provides a user interpretable designation for the level of completion of the action.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

assigned_request: the **versioned_action_request** to which the status applies.

10.4.8 action_resource

An **action_resource** is a thing that is identified as being needed to carry out an action.

EXPRESS specification

```

*)
ENTITY action_resource;
  name      : label;
  description : OPTIONAL text;
  usage     : SET [1:?] OF supported_item;
  kind      : action_resource_type;
END_ENTITY; -- action_resource
(*

```

Attribute definitions

name: the **label** by which the **action_resource** is known.

description: the **text** that characterizes the **action_resource**. The value of the attribute need not be specified.

usage: a set of **supported_items** for which the **action_resource** is used.

kind: the **action_resource_type** that specifies the sort of **action_resource** that is being used.

10.4.9 action_resource_relationship

An **action_resource_relationship** relates two instances of the entity data type **action_resource** and provides an identification and description of this relationship.

NOTE 1 The role of **action_resource_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 This entity, together with the **action_resource** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY action_resource_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_resource : action_resource;
  related_resource : action_resource;
END_ENTITY; -- action_resource_relationship
(*
```

Attribute definitions

name: the **label** by which the **action_resource_relationship** is known.

description: the **text** that characterizes the **action_resource_relationship**. The value of the attribute need not be specified.

relating_resource: one of the instances of **action_resource** that is a part of the relationship.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_resource: the other instance of **action_resource** that is a part of the relationship. If one element of the relationship is dependent upon the other this attribute shall be the dependent one.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

10.4.10 **action_resource_type**

An **action_resource_type** is the identification of the kind of **action_resource** needed to carry out an action.

EXAMPLE This entity may be used to specify the kind of tool needed to perform an process operation.

EXPRESS specification

```
*)
ENTITY action_resource_type;
  name : label;
END_ENTITY; -- action_resource_type
(*
```

Attribute definitions

name: the **label** by which the **action_resource_type** is known.

10.4.11 **action_status**

An **action_status** is the association of a status with an **executed_action**.

NOTE 1 Information about the date and time may be associated with the **action_status** through the use of **date_assignment**, **date_and_time_assignment**, or **time_assignment**.

EXPRESS specification

```

*)
ENTITY action_status;
  status          : label;
  assigned_action : executed_action;
END_ENTITY; -- action_status
( *

```

Attribute definitions

status: the **label** that provides a user interpretable designation for the level of completion of the action.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE Terms such as 'pending', 'completed', or 'scheduled' are examples of status.

assigned_action: the **executed_action** to which the status applies.

10.4.12 directed_action

A **directed_action** is a type of **executed_action** that is governed by an **action_directive**.

EXAMPLE A **directed_action** could be the inspection of a building as directed by city officials according to the city building codes for earthquake safety. The action is the inspection of the building. The directive is issued by city officials guided by the city building codes. In an application protocol, the building authority may be associated with an **organization_assignment**. The building codes may be associated with a **document_reference**.

EXPRESS specification

```

*)
ENTITY directed_action
  SUBTYPE OF (executed_action);
  directive : action_directive;
END_ENTITY; -- directed_action
( *

```

Attribute definitions

directive: the **action_directive** that governs the **directed_action**.

10.4.13 executed_action

An **executed_action** is a type of **action** that is either completed, partially completed or just identified. It may but need not have status information associated with it.

NOTE 1 The role of **executed_action** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Status information is associated to **executed_action** through **action_status**.

EXAMPLE An **executed_action** could be to 'paint the office' with a status of 'scheduled'. The action is 'paint the office'. The status further qualifies the action as 'planned', 'scheduled', or 'completed'.

EXPRESS specification

```
*)
ENTITY executed_action
  SUBTYPE OF (action);
END_ENTITY; -- executed_action
(*
```

10.4.14 versioned_action_request

A **versioned_action_request** specifies a desired result.

NOTE The desired result being identified and described may be obtained through one of more **action_methods**.

EXPRESS specification

```
*)
ENTITY versioned_action_request;
  id          : identifier;
  version     : label;
  purpose     : text;
  description : OPTIONAL text;
END_ENTITY; -- versioned_action_request
(*
```

id: the **identifier** that distinguishes the **versioned_action_request**.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 2 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

version: the identification of the version of the **versioned_action_request**.

purpose: an informal description of the reason for the **versioned_action_request**.

description: the **text** that characterizes the **versioned_action_request**. The value of the attribute need not be specified.

10.5 Action function definitions

10.5.1 acyclic_action_method_relationship

The **acyclic_action_method_relationship** function determines whether the graph of instances of the entity data type **action_method** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **action_method_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **action_method_relationship** entity include rules that use this function.

EXPRESS specification

```

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

  LOCAL
    x          : SET OF action_method_relationship;
  END_LOCAL;

  IF relation.relying_method IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY (am <* bag_to_set
             (USEDIN (relation.relying_method,
                    'ACTION_SCHEMA.' +
                    'ACTION_METHOD_RELATIONSHIP.' +
                    'RELATED_METHOD')) |
             specific_relation IN TYPEOF (am));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_action_method_relationship
      (x[i],
       relatives + relation.relying_method,
       specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
  RETURN(TRUE);
END_FUNCTION; -- acyclic_action_method_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **action_method** that the function is searching for in the relating_action_method parameter of the relation argument.

specific_relation: (input) the fully qualified name of a subtype of the **action_method_relationship** entity.

10.5.2 acyclic_action_relationship

The **acyclic_action_relationship** function determines whether the graph of instances of the entity data type **action** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **action_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **action_relationship** entity include rules that use this function.

EXPRESS specification

```

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

LOCAL
  x      : SET OF action_relationship;
END_LOCAL;

IF relation.relating_action IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (actn <* bag_to_set
            (USEDIN (relation.relating_action,
                    'ACTION_SCHEMA.' +
                    'ACTION_RELATIONSHIP.' +
                    'RELATED_ACTION')) |
            specific_relation IN TYPEOF (actn));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_action_relationship
    (x[i],
     relatives + relation.relating_action,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_action_relationship
(*

```

Argument definitions:

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

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

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

10.5.3 **acyclic_action_resource_relationship**

The **acyclic_action_resource_relationship** function determines whether the graph of instances of the entity data type **action_resource** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **action_resource_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **action_resource_relationship** entity include rules that use this function.

EXPRESS specification

```

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

LOCAL
  x      : SET OF action_resource_relationship;
END_LOCAL;

IF relation.relating_resource IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (ar <* bag_to_set
            (USEDIN (relation.relating_resource,
                    'ACTION_SCHEMA.' +
                    'ACTION_RESOURCE_RELATIONSHIP.' +
                    'RELATED_RESOURCE')) |
            specific_relation IN TYPEOF (ar));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_action_resource_relationship
    (x[i],
     relatives + relation.relating_resource,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_action_resource_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **action_resource** that the function is searching for in the relating **action_resource** parameter of the relation argument.

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

EXPRESS specification

```

*)
END_SCHEMA; -- action_schema
(*

```

11 Certification

The following EXPRESS declaration begins the **certification_schema** and identifies the necessary external references.

EXPRESS specification

```
*)  
SCHEMA certification_schema;  
  
REFERENCE FROM support_resource_schema  
  (label,  
   text);  
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

support_resource_schema clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 This schema contains support resources.

11.1 Introduction

The subject of the **certification_schema** is the definition of certifications. A certification assures and validates product data.

EXAMPLE 1 A material certificate states the chemical composition of one or more physical pieces of material, for example, purchased raw material. The presence of the material certificate removes the need to test the composition of the material; it allows the specified material composition to be accepted as fact without further investigation.

EXAMPLE 2 A certified supplier can supply goods that do not require checking.

11.2 Fundamental concepts and assumptions

Certification information can be attached to any aspect of product data.

11.3 Certification entity definitions

11.3.1 certification

A **certification** is documentation that asserts facts.

NOTE The role of **certification** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXPRESS specification

```
*)  
ENTITY certification;  
  name      : label;  
  purpose   : text;  
  kind      : certification_type;  
END_ENTITY; -- certification  
(*
```

Attribute definitions

name: the **label** by which the **certification** is known.

purpose: an informal description of the reason why the **certification** is applied.

EXAMPLE A purpose for material certification could be the fact that certain customers demand material certification for the products that are supplied to them.

kind: the **certification_type** that defines the class of **certification** that is applied.

11.3.2 certification_type

A **certification_type** is the kind of **certification** granted.

EXAMPLE 1 Suppliers and manufacturers can be certified.

EXPRESS specification

```
*)
ENTITY certification_type;
  description : label;
END_ENTITY; -- certification_type
(*
```

Attribute definitions

description: the **text** that characterizes the **certification_type**.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE 2 The value of this attribute could be 'supplier' or 'manufacturer'.

EXPRESS specification

```
*)
END_SCHEMA; -- certification_schema
(*
```

12 Approval

The following EXPRESS declaration begins the **approval_schema** and identifies the necessary external references.

EXPRESS specification

```
*)
SCHEMA approval_schema;

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   get_description_value,
   get_role,
   object_role,
   -- ISO 10303-41
```

```
role_association);  
  
REFERENCE FROM date_time_schema -- ISO 10303-41  
(date_time_select);  
  
REFERENCE FROM person_organization_schema -- ISO 10303-41  
(person_organization_select);  
  
REFERENCE FROM support_resource_schema -- ISO 10303-41  
(bag_to_set,  
label,  
text);  
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
date_time_schema	clause 16 of this part of ISO 10303
person_organization_schema	clause 15 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

12.1 Introduction

The subject of the **approval_schema** is the description of authorization data. Approval information concerns the acceptance of product data.

EXAMPLE One version of a product may be approved for manufacture whilst another may still be undergoing design.

12.2 Fundamental concepts and assumptions

Approval information can be attached to any aspect of product data.

12.3 Approval entity definitions

12.3.1 approval

An **approval** is a confirmation of the quality of the product data that it is related to.

NOTE 1 The role of **approval** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXPRESS specification

```

*)
ENTITY approval;
  status : approval_status;
  level : label;
END_ENTITY; -- approval
(*

```

Attribute definitions

status: the **label** that provides a user interpretable designation for the level of completion of the action.

level: the type or level of **approval** in terms of its usage.

NOTE 2 This usage may be implied rather than explicit.

EXAMPLE One possible level of **approval** is 'released for production'; this explicitly identifies the approved usage. Another possible level is 'preliminary design completed'; this only implies the approved usage that will depend upon company-specific procedures.

12.3.2 approval_date_time

An **approval_date_time** is an association between an **approval** and a date, time, or date and time.

EXPRESS specification

```

*)
ENTITY approval_date_time;
  date_time      : date_time_select;
  dated_approval : approval;
DERIVE
  role           : object_role := get_role (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
  'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- approval_date_time
(*

```

Attribute definitions

date_time: the moment when the dated_approval is given.

dated_approval: the **approval** with which the date or time is associated.

role: the **object_role** that specifies the purpose of the association of the **approval_date_time** with product data.

NOTE 1 This attribute is an upwardly compatible addition to **approval_date_time** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **approval_date_time** shall be the item_with_role in at most one **role_association**.

NOTE 2 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

12.3.3 approval_person_organization

An **approval_person_organization** is an association between an **approval** and a given person, organization or person and organization.

EXPRESS specification

```
*)
ENTITY approval_person_organization;
  person_organization : person_organization_select;
  authorized_approval : approval;
  role                 : approval_role;
END_ENTITY; -- approval_person_organization
(*
```

Attribute definitions

person_organization: the person or organization playing the given role.

authorized_approval: the **approval** that is effected by the person or organization.

role: the **approval_role** that specifies the purpose of the association of the **approval** with a person or organization.

12.3.4 approval_relationship

An **approval_relationship** relates two instances of the entity data type **approval** with a description of their relationship.

NOTE 1 The role of **approval_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 This entity, together with the **approval** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY approval_relationship;
  name           : label;
  description     : OPTIONAL text;
  relating_approval : approval;
  related_approval  : approval;
END_ENTITY; -- approval_relationship
(*
```

Attribute definitions

name: the **label** by which the **approval_relationship** is known.

description: the **text** that characterizes the **approval_relationship**. The value of the attribute need not be specified.

relating_approval: one of the instances of **approval** that is a part of the relationship.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_approval: the other instance of **approval** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

12.3.5 approval_role

An **approval_role** defines a function performed by a **person_organization** with respect to an **approval**.

EXPRESS specification

```
* )
ENTITY approval_role;
  role          : label;
DERIVE
  description   : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- approval_role
(*
```

Attribute definitions

role: the **label** by which the **approval_role** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **approval_role**.

NOTE 2 This attribute is an upwardly compatible addition to **approval_role** as specified in ISO 10303-41:1994.

Formal proposition

WR1: Each **approval_role** shall be the **described_item** in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

12.3.6 approval_status

An **approval_status** is the ranking that specifies the state of an **approval**.

EXAMPLE 'Approved' and 'disapproved' are examples of **approval_status**.

EXPRESS specification

```

*)
ENTITY approval_status;
  name : label;
END_ENTITY; -- approval_status
( *

```

Attribute definitions

name: the **label** by which the **approval_status** is known.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

12.4 Approval function definition: **acyclic_approval_relationship**

The **acyclic_approval_relationship** function determines whether the graph of instances of the entity data type **approval** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **approval_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **approval_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_approval_relationship
  (relation      : approval_relationship;
   relatives     : SET [1:?] OF approval;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x      : SET OF approval_relationship;
END_LOCAL;

IF relation.relatng_approval IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (app <* bag_to_set
            (USEDIN (relation.relatng_approval,
                    'APPROVAL_SCHEMA.' +
                    'APPROVAL_RELATIONSHIP.' +
                    'RELATED_APPROVAL')) |
            specific_relation IN TYPEOF (app));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_approval_relationship
    (x[i],
     relatives + relation.relatng_approval,
     specific_relation) THEN
    RETURN (FALSE);
  END_IF;

```

```

    END_REPEAT;
    RETURN(TRUE);
END_FUNCTION; -- acyclic_approval_relationship
(*)

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **approval** for which the function is searching in the relating_approval parameter of the relation argument.

specific_relation: (input) the fully qualified name of a subtype of the **approval_relationship** entity.

EXPRESS specification

```

*)
END_SCHEMA; -- approval_schema
(*)

```

13 Contract

The following EXPRESS declaration begins the **contract_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA contract_schema;

REFERENCE FROM support_resource_schema -- ISO 10303-41
    (bag_to_set,
     identifier,
     label,
     text);
(*)

```

NOTE 1 The schema referenced above is specified in the following part of ISO 10303:

support_resource_schema clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

13.1 Introduction

The subject of the **contract_schema** is the description of contract agreements.

13.2 Fundamental concepts and assumptions

Contracts are binding agreements. Contract information may be attached to any aspect of a product data.

13.3 Contract entity definitions

13.3.1 contract

A **contract** is a binding agreement.

NOTE Contracts may be enforceable by law.

EXPRESS specification

```
*)
ENTITY contract;
  name      : label;
  purpose   : text;
  kind      : contract_type;
END_ENTITY; -- contract
(*
```

Attribute definitions

name: the **label** by which the **contract** is known.

purpose: an informal description of the reason for the **contract**.

kind: the **contract**'s type.

13.3.2 contract_relationship

A **contract_relationship** relates two instances of the entity data type **contract**.

NOTE 1 The role of **contract_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **contract** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY contract_relationship;
  id          : identifier;
  name        : label;
  description  : OPTIONAL text;
  relating_contract : contract;
  related_contract : contract;
END_ENTITY; -- contract_relationship
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **contract_relationship**.

NOTE 4 The context in which id is used as a discriminating characteristic can be identified in an annotated EXPRESS schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **contract_relationship** is known.

description: the **text** that characterizes the **contract_relationship**. The value of the attribute need not be specified.

relating_contract: one of the instances of **contract** that is a part of the relationship.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_contract: the other instance of **contract** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 6 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

13.3.3 contract_type

A **contract_type** is the kind of information that an instance of **contract** conveys.

EXAMPLE A contract for an organization that is supplying goods to another organization could be 'fixed' or 'cost plus'.

EXPRESS specification

```
*)
ENTITY contract_type;
  description : label;
END_ENTITY; -- contract_type
(*
```

Attribute definitions

description: the **text** that characterizes the **contract_type**. The value of the attribute need not be specified.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

13.4 Contract function definition: acyclic_contract_relationship

The **acyclic_contract_relationship** function determines whether the graph of instances of the entity data type **contract** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **contract_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **contract_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_contract_relationship
  (relation          : contract_relationship;
   relatives        : SET [1:?] OF contract;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x          : SET OF contract_relationship;
END_LOCAL;

IF relation.relatng_contract IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (cont <* bag_to_set
            (USEDIN (relation.relatng_contract,
                    'CONTRACT_SCHEMA.' +
                    'CONTRACT_RELATIONSHIP.' +
                    'RELATED_CONTRACT')) |
            specific_relation IN TYPEOF (cont));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_contract_relationship
    (x[i],
     relatives + relation.relatng_contract,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_contract_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **contract** for which the function is searching in the **relatng_contract** parameter of the relation argument.

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

EXPRESS specification

```

*)
END_SCHEMA; -- contract_schema
(*

```

14 Security classification

The following EXPRESS declaration begins the **security_classification_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA security_classification_schema;

REFERENCE FROM support_resource_schema
  (label,
   text);
(*

```

NOTE 1 The schema referenced above is specified in the following part of ISO 10303:

support_resource_schema clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

14.1 Introduction

The subject of the **security_classification_schema** is the definition of degrees of secrecy to be specified.

14.2 Fundamental concepts and assumptions

A security classification is the level of confidentiality that is required for the purpose of product data protection. Security classifications are assigned by some authoritative agency. Security classification information can be attached to any aspect of product data.

14.3 Security classification entity definitions

14.3.1 security_classification

A **security_classification** is the level of confidentiality that is required for the purpose of product data protection.

NOTE The role of can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXPRESS specification

```

*)
ENTITY security_classification;
  name          : label;
  purpose       : text;
  security_level : security_classification_level;
END_ENTITY; -- security_classification
(*

```

Attribute definitions

name: the **label** by which the **security_classification** is known.

EXAMPLE 'National Security' is a **security_classification_role**. **Security_classification_assignment_role** assigns the **security_classification_role** to a **security_classification_assignment**.

purpose: an informal description of the intent of the **security_classification**.

security_level: the **security_classification_level** that specifies the required degree of security.

14.3.2 security_classification_level

A **security_classification_level** is a category of security required for product data protection.

EXAMPLE 'Confidential', 'secret', and 'top secret' are examples of **security_classification_level** levels.

EXPRESS specification

```
*)
ENTITY security_classification_level;
  name : label;
END_ENTITY; -- security_classification_level
(*
```

Attribute definitions

name: the **label** by which the **security_classification_level** is known.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXPRESS specification

```
*)
END_SCHEMA; -- security_classification_schema
(*
```

15 Person organization

The following EXPRESS declaration begins the **person_organization_schema** and identifies the necessary external references.

EXPRESS specification

```
*)
SCHEMA person_organization_schema;

REFERENCE FROM basic_attribute_schema -- ISO 10303-41
  (description_attribute,
   get_description_value,
   get_id_value,
   get_name_value,
   get_role,
   id_attribute,
```

```

name_attribute,
object_role,
role_association);

```

```

REFERENCE FROM support_resource_schema          -- ISO 10303-41
(bag_to_set,
 identifier,
 label,
 text);
(*)

```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

15.1 Introduction

The subject of the **person_organization_schema** is the description of information that identifies people and organizations.

15.2 Fundamental concepts and assumptions

People and organizations are associated with product data to provide data concerning administrative structures and individuals who serve as points of contact or have particular responsibility in those structures. People and organizations may be associated with any aspect of product data.

15.3 Person organization type definition: person_organization_select

The **person_organization_select** type allows for the designation of a **person**, an **organization** or a **person_and_organization**.

EXPRESS specification

```

*)
TYPE person_organization_select = SELECT
  (person,
   organization,
   person_and_organization);
END_TYPE; -- person_organization_select
(*)

```

15.4 Person organization entity definitions

15.4.1 address

An **address** is the information necessary for communicating, using one or more communication methods

EXPRESS specification

```

*)
ENTITY address;
  internal_location      : OPTIONAL label;
  street_number         : OPTIONAL label;
  street                : OPTIONAL label;
  postal_box            : OPTIONAL label;
  town                  : OPTIONAL label;
  region                : OPTIONAL label;
  postal_code           : OPTIONAL label;
  country                : OPTIONAL label;
  facsimile_number      : OPTIONAL label;
  telephone_number     : OPTIONAL label;
  electronic_mail_address : OPTIONAL label;
  telex_number          : OPTIONAL label;
DERIVE
  name                  : label := get_name_value (SELF);
  url                   : identifier := get_id_value (SELF);
WHERE
  WR1: EXISTS(internal_location)      OR
        EXISTS(street_number)         OR
        EXISTS(street)                OR
        EXISTS(postal_box)            OR
        EXISTS(town)                  OR
        EXISTS(region)                OR
        EXISTS(postal_code)           OR
        EXISTS(country)               OR
        EXISTS(facsimile_number)      OR
        EXISTS(telephone_number)     OR
        EXISTS(electronic_mail_address) OR
        EXISTS(telex_number);
END_ENTITY; -- address
( *

```

Attribute definitions

internal_location: an organization-defined address for internal mail delivery.

street_number: the number of a location on a street.

street: the name of a street.

postal_box: the number of a postal box.

town: the name of a town.

region: the name of a region.

EXAMPLE The counties of Great Britain and the states of the United States of America are examples of regions.

postal_code: the code that is used by the **country**'s postal service.

country: the name of a country.

facsimile_number: the number at which facsimiles may be received.

telephone_number: the number at which telephone calls may be received.

electronic_mail_address: the electronic address at which electronic mail may be received.

telex_number: the number at which telex messages may be received.

name: the **label** that specifies the name of the location identified by the **address**.

EXAMPLE "White House", "Buckingham Palace", "Kremlin", "World Trade Center"

url: the text that specifies the uniform resource locator associated with the address.

NOTE The format of uniform resource locators is defined in IETF RFC 1738 [4].

EXAMPLE URLs are used to identify world wide web sites, file transfer protocol (FTP) servers, etc.

Formal propositions:

WR1: At least one of the attributes shall have a value.

NOTE A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

15.4.2 organization

An **organization** is an administrative structure.

EXPRESS specification

```
*)
ENTITY organization;
  id          : OPTIONAL identifier;
  name       : label;
  description : OPTIONAL text;
END_ENTITY; -- organization
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **organization**. The value of this attribute need not be specified.

NOTE The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **organization** is known.

description: the **text** that characterizes the **organization**. The value of the attribute need not be specified.

15.4.3 organization_relationship

An **organization_relationship** relates two instances of the entity data type **organization** with a description of their relationship.

EXAMPLE Each department of an enterprise, and the enterprise itself, may be regarded as individual organizations. The fact that the departments are a part of the whole enterprise may be represented using this entity.

NOTE 1 The role of **organization_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **organization** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY organization_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_organization : organization;
  related_organization : organization;
END_ENTITY; -- organization_relationship
(*
```

Attribute definitions

name: the **label** by which the **organization_relationship** is known.

description: the **text** that characterizes the **organization_relationship**. The value of the attribute need not be specified.

relating_organization: one of the instances of **organization** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_organization: the other instance of **organization** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

15.4.4 organization_role

An **organization_role** defines a role for an **organization_assignment** and a description of that role.

EXPRESS specification

```

*)
ENTITY organization_role;
  name      : label;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- organization_role
(*

```

Attribute definitions

name: the **label** by which the **organization_role** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **organization_role**. The value of the attribute need not be specified.

NOTE 2 This attribute is an upwardly compatible addition to **organization_role** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **organization_role** shall be the **described_item** in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

15.4.5 organizational_address

An **organizational_address** is an **address** for one or more organizations.

EXPRESS specification

```

*)
ENTITY organizational_address
  SUBTYPE OF (address);
  organizations : SET [1:?] OF organization;
  description   : OPTIONAL text;
END_ENTITY; -- organizational_address
(*

```

Attribute definitions

organizations: the organizations located at the **address**.

description: the **text** that characterizes the **organizational_address**. The value of the attribute need not be specified.

15.4.6 organizational_project

An **organizational_project** is a project for which one or more **organization** entity data types are responsible.

EXPRESS specification

```

*)
ENTITY organizational_project;
  name                : label;
  description          : OPTIONAL text;
  responsible_organizations : SET[1:?] OF organization;
DERIVE
  id                  : identifier := get_id_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                      'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- organizational_project
(*

```

Attribute definitions

name: the **label** by which the **organizational_project** is known.

description: the **text** that characterizes the **organizational_project**. The value of the attribute need not be specified.

responsible_organizations: the **organization** entity data types indicating who has project oversight.

id: the **identifier** that distinguishes the **organizational_project**.

NOTE 1 This attribute is an upwardly compatible addition to **organizational_project** as specified in ISO 10303-41:1994.

NOTE 2 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

Formal propositions:

WR1: Each **organizational_project** shall be the **identified_item** in at most one **id_attribute**.

NOTE 3 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

15.4.7 organizational_project_relationship

An **organizational_project_relationship** relates two instances of the entity data type **organizational_project** with an identification and description of their relationship.

NOTE 1 The role of **organizational_project_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **organizational_project** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY organizational_project_relationship;
  name                : label;
  description          : OPTIONAL text;
  relating_organizational_project : organizational_project;
  related_organizational_project  : organizational_project;
END_ENTITY; -- organizational_project_relationship
(*
```

Attribute definitions

name: the **label** by which the **organizational_project_relationship** is known.

description: the **text** that characterizes the **organizational_project_relationship**. The value of the attribute need not be specified.

relating_organizational_project: one of the instances of **organizational_project** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_organizational_project: the other instance of **organizational_project** that is a part of the relationship. If one element of the relationship is dependent up on the other, this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

15.4.8 person

A **person** is an individual human being.

EXPRESS specification

```
*)
ENTITY person;
  id                : identifier;
  last_name         : OPTIONAL label;
  first_name        : OPTIONAL label;
  middle_names      : OPTIONAL LIST [1:?] OF label;
  prefix_titles     : OPTIONAL LIST [1:?] OF label;
  suffix_titles     : OPTIONAL LIST [1:?] OF label;
WHERE
  WR1: EXISTS(last_name) OR EXISTS(first_name);
END_ENTITY; -- person
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **person**.

NOTE 1 The identification of a person is usually context dependent.

NOTE 2 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

EXAMPLE In the USA a person's **id** would be his or her social security number whereas in the United Kingdom it would be his or her national insurance number.

NOTE 3 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

last_name: the **person's** surname.

first_name: the first element of the **person's** list of forenames.

middle_names: the **person's** other forenames, if there are any.

prefix_titles: the **text** that specifies the **person's** social or professional standing and appear before his or her names.

suffix_titles: the **text** that specifies the **person's** social or professional standing and appear after his or her names.

Formal propositions:

WR1: Either the last name, first name or the last and first name shall be defined.

15.4.9 **person_and_organization**

A **person_and_organization** is a person in an organization.

EXPRESS specification

```
*)
ENTITY person_and_organization;
  the_person      : person;
  the_organization : organization;
DERIVE
  name           : label := get_name_value (SELF);
  description    : text  := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- person_and_organization
(*
```

Attribute definitions

the_person: the instance of the **person** entity data type who is related to an instance of **organization**.

the_organization: the instance of the **organization** entity data type to which the instance of **person** is related.

name: the **label** by which the **person_and_organization** is known.

NOTE 1 This attribute is an upwardly compatible addition to **person_and_organization** as specified in ISO 10303-41:1994.

description: the **text** that characterizes the **person_and_organization**.

NOTE 2 This attribute is an upwardly compatible addition to **person_and_organization** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **person_and_organization** shall be the **described_item** in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

WR2: Each **person_and_organization** shall be the **named_item** in at most one **name_attribute**.

NOTE 4 The **name_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 5 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

15.4.10 **person_and_organization_role**

A **person_and_organization_role** defines a role for a **person_and_organization_assignment** and a description of that role.

EXAMPLE The role of a **person** could be 'buyer' in the context of the **organization** where he/she works and it could be 'customer' in the context of the **organization** from which he/she purchases goods.

EXPRESS specification

```

*)
ENTITY person_and_organization_role;
  name          : label;
DERIVE
  description   : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                      'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- person_and_organization_role
(*)

```

Attribute definitions

name: the **label** by which the **person_and_organization_role** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **person_and_organization_role**.

NOTE 2 This attribute is an upwardly compatible addition to **person_and_organization_role** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **person_and_organization_role** shall be the **described_item** in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

15.4.11 person_role

A **person_role** defines a role for a **person_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY person_role;
  name          : label;
DERIVE
  description   : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- person_role
(*
```

Attribute definitions

name: the **label** by which the **person_role** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **person_role**.

NOTE 2 This attribute is an upwardly compatible addition to **person_role** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **person_role** shall be the **described_item** in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

15.4.12 personal_address

A **personal_address** is an **address** for one or more persons.

EXPRESS specification

```
*)
ENTITY personal_address
  SUBTYPE OF (address);
  people      : SET [1:?] OF person;
  description : OPTIONAL text;
END_ENTITY; -- personal_address
(*
```

Attribute definitions

people: the people who reside at the **address**.

description: the **text** that characterizes the **personal_address**. The value of the attribute need not be specified.

15.5 Person organization function definitions

15.5.1 acyclic_organization_relationship

The **acyclic_organization_relationship** function determines whether the graph of instances of the entity data type **organization** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **organization_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **organization_relationship** entity include rules that use this function.

EXPRESS specification

```
*)
FUNCTION acyclic_organization_relationship
  (relation      : organization_relationship;
   relatives     : SET [1:?] OF organization;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x          : SET OF organization_relationship;
END_LOCAL;

IF relation.relativing_organization IN relatives THEN
  RETURN (FALSE);
END_IF;
```

```

x := QUERY (org <* bag_to_set
            (USEDIN (relation.relativing_organization,
                    'PERSON_ORGANIZATION_SCHEMA.' +
                    'ORGANIZATION_RELATIONSHIP.' +
                    'RELATED_ORGANIZATION')) |
            specific_relation IN TYPEOF (org));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_organization_relationship
    (x[i],
     relatives + relation.relativing_organization,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_organization_relationship
(*)

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **organization** for which the function is searching in the **relating_organization** attribute of the relation argument.

specific_relation: (input) the fully qualified name of a subtype of the **organization_relationship** entity.

15.5.2 acyclic_organizational_project_relationship

The **acyclic_organizational_project_relationship** function determines whether the graph of instances of the entity data type **organizational_project** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **organizational_project_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **organizational_project_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_organizational_project_relationship
  (relation      : organizational_project_relationship;
   relatives     : SET [1:?] OF organizational_project;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x      : SET OF organizational_project_relationship;
END_LOCAL;

IF relation.relativing_organizational_project IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (op <* bag_to_set
            (USEDIN (relation.relativing_organizational_project,

```

```

        'PERSON_ORGANIZATION_SCHEMA.' +
        'ORGANIZATIONAL_PROJECT_RELATIONSHIP.' +
        'RELATED_ORGANIZATIONAL_PROJECT')) |
        specific_relation IN TYPEOF (op));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_organizational_project_relationship
    (x[i],
     relatives + relation.relativing_organizational_project,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_organizational_project_relationship
(*)

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **organizational_project** for which the function is searching in the **relating_organizational_project** attribute of the **relation** argument.

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

EXPRESS specification

```

*)
END_SCHEMA; -- person_organization_schema
(*)

```

16 Date time

The following EXPRESS declaration begins the **date_time_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA date_time_schema;

REFERENCE FROM basic_attribute_schema          -- ISO 10303-41
  (description_attribute,
   get_description_value);

REFERENCE FROM measure_schema                  -- ISO 10303-41
  (time_measure_with_unit);

REFERENCE FROM support_resource_schema        -- ISO 10303-41
  (bag_to_set,
   identifier,
   label,
   text);
(*)

```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
measure_schema	clause 21 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

16.1 Introduction

The subject of the **date_time_schema** is the definition of dates, both calendar and ordinal, time of day, combinations of date and time of day, and periods of time.

16.2 Fundamental concepts and assumptions

Any kind of product data may have a date, time, or date and time associated with it.

16.3 Date time type definitions

16.3.1 ahead_or_behind

The **ahead_or_behind** type is used to specify whether a given time is ahead of or behind coordinated universal time.

NOTE Coordinated Universal Time (UTC) is the international time standard. It is the current term for what was commonly referred to as Greenwich Meridian Time (GMT). Zero (0) hours UTC is midnight in Greenwich England, which lies on the zero longitudinal meridian. Coordinated universal time is based on a 24 hour clock; therefore, afternoon hours such as 4 pm are expressed as 16:00 UTC (sixteen hours, zero minutes).

EXPRESS specification

```
*)
TYPE ahead_or_behind = ENUMERATION OF
  (ahead,
   exact,
   behind);
END_TYPE; -- ahead_or_behind
(*
```

16.3.2 date_time_or_event_occurrence

The **date_time_or_event** type allows for the designation of a **date_time_select** including **date** and **date_and_time** or an **event_occurrence**.

EXPRESS specification

```
*)
TYPE date_time_or_event_occurrence = SELECT
  (date_time_select,
   event_occurrence);
```

```
END_TYPE; -- date_time_or_event_occurrence
(*
```

16.3.3 date_time_select

A **date_time_select** type allows for the designation of a **date**, **local_time**, or a **date_and_time**.

EXPRESS specification

```
*)
TYPE date_time_select = SELECT
  (date,
   local_time,
   date_and_time);
END_TYPE; -- date_time_select
(*
```

16.3.4 day_in_month_number

A **day_in_month_number** is the position of the specified day in a month.

EXPRESS specification

```
*)
TYPE day_in_month_number = INTEGER;
WHERE
  WR1: {1 <= SELF <= 31};
END_TYPE; -- day_in_month_number
(*
```

16.3.5 day_in_week_number

A **day_in_week_number** is the value of day as defined in ISO 8601 (subclause 5.2.3).

NOTE Monday is day number 1, Tuesday is day number 2, Wednesday is day number 3, Thursday is day number 4, Friday is day number 5, Saturday is day number 6, and Sunday is day number 7.

EXPRESS specification

```
*)
TYPE day_in_week_number = INTEGER;
WHERE
  WR1: {1 <= SELF <= 7 };
END_TYPE; -- day_in_week_number
(*
```

Formal propositions:

WR1: The value of the integer shall be between 1 and 7.

16.3.6 day_in_year_number

A **day_in_year_number** is the position of the specified day in a year.

EXAMPLE The 27th day of March is day 86 in years that are not leap years and day 87 in leap years.

EXPRESS specification

```
*)
TYPE day_in_year_number = INTEGER;
WHERE
  WR1: { 1 <= SELF <= 366 };
END_TYPE; -- day_in_year_number
(*
```

16.3.7 hour_in_day

An **hour_in_day** is the hour element of a specified time on a 24 hour clock. Midnight shall be represented by the value zero.

EXAMPLE The hour_in_day corresponding to 3 o'clock in the afternoon is 15.

NOTE Although ISO 8601 allows two representations for midnight, 0000 and 2400, this part of ISO 10303 restricts the representation to the first value.

EXPRESS specification

```
*)
TYPE hour_in_day = INTEGER;
WHERE
  WR1: { 0 <= SELF < 24 };
END_TYPE; -- hour_in_day
(*
```

Formal propositions:

WR1: The value of the integer shall be between 0 and 23.

16.3.8 minute_in_hour

A **minute_in_hour** is the minute element of a specified time.

EXPRESS specification

```
*)
TYPE minute_in_hour = INTEGER;
WHERE
  WR1: { 0 <= SELF <= 59 };
END_TYPE; -- minute_in_hour
(*
```

Formal propositions:

WR1: The value of the integer shall be between 0 and 59.

16.3.9 month_in_year_number

A **month_in_year_number** is the position of the specified month in a year as defined in ISO 8601 (subclause 5.2.1).

NOTE January is month number 1, February is month number 2, March is month number 3, April is month number 4, May is month number 5, June is month number 6, July is month number 7, August is month number 8,

September is month number 9, October is month number 10, November is month number 11, and December is month number 12.

EXPRESS specification

```
*)
TYPE month_in_year_number = INTEGER;
WHERE
  WR1: { 1 <= SELF <= 12 };
END_TYPE; -- month_in_year_number
(*
```

Formal propositions:

WR1: The value of the integer shall be between 1 and 12.

16.3.10 second_in_minute

A **second_in_minute** is the second element of a specified time.

EXPRESS specification

```
*)
TYPE second_in_minute = REAL;
WHERE
  WR1: { 0 <= SELF <= 60.0 };
END_TYPE; -- second_in_minute
(*
```

Formal propositions:

WR1: The value of the real number shall be between 0 to 60.0.

NOTE 1 A value of 60 allows for leap seconds.

NOTE 2 The mean solar time is determined by the rotation of the earth. Leap seconds are added as required, usually in the middle or at the end of a year, and ensure that the legal time does not differ from the nonuniform mean solar time by more than one second, in spite of the variations of the earth rotation.

16.3.11 week_in_year_number

A **week_in_year_number** is the value of the calendar week as defined in ISO 8601 (subclause 3.1.7).

NOTE Week number 1 is the week containing the first Thursday of the year. This is equivalent to saying that week number 1 contains the date 4th January.

EXPRESS specification

```
*)
TYPE week_in_year_number = INTEGER;
WHERE
  WR1: { 1 <= SELF <= 53 };
END_TYPE; -- week_in_year_number
(*
```

Formal propositions:

WR1: The value of the integer shall be between 1 and 53.

16.3.12 year_number

A **year_number** is the year as defined in the Gregorian calendar. The **year_number** shall be completely and explicitly specified using as many digits as necessary to unambiguously convey the century and year within the century. Truncated year numbers shall not be used.

NOTE ISO 8601:1988 defines the Gregorian calendar.

EXAMPLE The year_number corresponding to the first manned landing on the moon is 1969 (not 69).

EXPRESS specification

```
*)
TYPE year_number = INTEGER;
END_TYPE; -- year_number
(*
```

16.4 Date time entity definitions

16.4.1 calendar_date

A **calendar_date** is a type of **date** defined as a day in a month of a year.

EXPRESS specification

```
*)
ENTITY calendar_date
  SUBTYPE OF (date);
  day_component : day_in_month_number;
  month_component : month_in_year_number;
WHERE
  WR1: valid_calendar_date (SELF);
END_ENTITY; -- calendar_date
(*
```

Attribute definitions

day_component: the day element of the **date**.

month_component: the month element of the **date**.

Formal propositions:

WR1: The entity shall define a valid calendar date.

NOTE If the month_component is 'April', 'June', 'August', or 'November' the day_component will be between 1 and 30; when the month_component is 'February' and the year_component is a leap year the day_component will be between 1 and 29; if the month_component is 'February' and the year_component is not a leap year the day_component will be between 1 and 28. Otherwise, the day_component will be between 1 and 31.

16.4.2 coordinated_universal_time_offset

A **coordinated_universal_time_offset** is the oriented offset (specified in hours and possibly minutes) from the coordinated universal time. The offset value shall be positive.

NOTE 1 Coordinated Universal Time (UTC) is the international time standard. It is the current term for what was commonly referred to as Greenwich Meridian Time (GMT). Zero (0) hours UTC is midnight in Greenwich England, which lies on the zero longitudinal meridian. Coordinated universal time is based on a 24 hour clock; therefore, afternoon hours such as 4 pm are expressed as 16:00 UTC (sixteen hours, zero minutes).

NOTE 2 A **coordinated_universal_time_offset** is used to relate a time to coordinated universal time by an offset (specified in hours and minutes) and a direction.

EXPRESS specification

```

*)
ENTITY coordinated_universal_time_offset;
  hour_offset      : INTEGER;
  minute_offset   : OPTIONAL INTEGER;
  sense           : ahead_or_behind;
DERIVE
  actual_minute_offset: INTEGER := NVL(minute_offset,0);
WHERE
  WR1: { 0 <= hour_offset < 24 };
  WR2: { 0 <= minute_offset <= 59 };
  WR3: NOT ((hour_offset <> 0) OR (actual_minute_offset <> 0))
        AND (sense = exact));
END_ENTITY; -- coordinated_universal_time_offset
(*)

```

Attribute definitions

hour_offset: the number of hours by which a time is offset from coordinated universal time.

minute_offset: the number of minutes by which a time is offset from coordinated universal time.

sense: the direction of the offset.

actual_minute_offset: the value of the number of minutes offset used to compute the **coordinated_universal_time_offset**, either the value of **minute_offset** or 0.

Formal propositions:

WR1: The **hour_offset** shall be a positive number, less than 24.

WR2: The **minute_offset** shall be a positive number, less than or equal to 59.

WR3: If the value of **sense** specifies that there is no offset from the Coordinated Universal time, **hour_offset** and **actual_minute_offset** shall both be equal to zero. If either **hour_offset** or **actual_minute_offset** is different from zero, the value of **sense** shall specify that there is an offset, either ahead or behind, from the Coordinated Universal time.

16.4.3 date

A **date** is the identification of a day or week in a year.

EXPRESS specification

```
*)
ENTITY date
  SUPERTYPE OF (ONEOF (calendar_date,
                       ordinal_date,
                       week_of_year_and_day_date));
  year_component : year_number;
END_ENTITY; -- date
(*
```

Attribute definitions

year_component: the year in which the **date** occurs.

16.4.4 date_and_time

A **date_and_time** is a moment of time on a particular day.

EXPRESS specification

```
*)
ENTITY date_and_time;
  date_component : date;
  time_component : local_time;
END_ENTITY; -- date_and_time
(*
```

Attribute definitions

date_component: the date element of the date time combination.

time_component: the time element of the date time combination.

16.4.5 date_role

A **date_role** defines a role for a **date_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY date_role;
  name : label;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- date_role
(*
```

Attribute definitions

name: the **label** by which the **date_role** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **date_role**. The value of the attribute need not be specified.

NOTE 2 This attribute is an upwardly compatible addition to **date_role** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **date_role** shall be the described_item in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

16.4.6 date_time_role

A **date_time_role** defines a role for a **date_and_time_assignment** and a description of that role.

EXPRESS specification

```
* )
ENTITY date_time_role;
  name          : label;
DERIVE
  description   : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- date_time_role
(*
```

Attribute definitions

name: the **label** by which the **date_time_role** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **date_time_role**.

NOTE 2 This attribute is an upwardly compatible addition to **date_time_role** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **date_time_role** shall be the described_item in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_-schema** is described in Annex E.

16.4.7 event_occurrence

An **event_occurrence** is the fact of an existence of a state at some point in time.

NOTE 1 The point in time of the existence may not be known in terms of a calendar date before the **event_occurrence** actually happens. One reason, why the **event_occurrence** cannot always be expressed as a calendar date, is that the **event_occurrence** is not possible to plan, for example a break down of a machine.

EXAMPLE 'Start of production' and 'break down of machine A' are examples for **event_occurrences**.

EXPRESS specification

```
*)
ENTITY event_occurrence;
  id      : identifier;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- event_occurrence
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **event_occurrence**.

NOTE 2 The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **event_occurrence** is known.

description: the **text** that characterizes the **event_occurrence**. The value of the attribute need not be specified.

16.4.8 event_occurrence_context_role

An **event_occurrence_context_assignment** is an association of an **event_occurrence_assignment** with product data representing the context for the assignment.

EXAMPLE For the **event_occurrence** 'start of production' the product, for which production starts, is the context for that **event_occurrence**.

EXPRESS specification

```
*)
ENTITY event_occurrence_context_role;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- event_occurrence_context_role
(*
```

Attribute definitions

name: the **label** by which the **event_occurrence_context_role** is known.

description: the **text** that characterizes the **event_occurrence_context_role**. The value of the attribute need not be specified.

16.4.9 event_occurrence_relationship

An **event_occurrence_relationship** relates two **event_occurrences** with a description of their relationship.

NOTE 1 The role of **event_occurrence_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 This entity, together with the **event_occurrence** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```

*)
ENTITY event_occurrence_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_event : event_occurrence;
  related_event  : event_occurrence;
END_ENTITY; -- event_occurrence_relationship
( *

```

Attribute definitions

name: the **label** by which the **event_occurrence_relationship** is known.

description: the **text** that characterizes the **event_occurrence_relationship**. The value of the attribute need not be specified.

relating_event: one of the instances of **event_occurrences** that is a part of the relationship.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_event: the other instance of **event_occurrence**. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

16.4.10 event_occurrence_role

An **event_occurrence_role** defines a role for an **event_occurrence_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY event_occurrence_role;
  name      : label;
  description : OPTIONAL text;
END_ENTITY; -- event_occurrence_role
(*
```

Attribute definitions

name: the **label** by which the **event_occurrence_role** is known.

description: the **text** that characterizes the **event_occurrence_role**. The value of the attribute need not be specified.

16.4.11 local_time

A **local_time** is an instance of time represented on a 24-hour clock by hour, minute, and second. The instance is expressed in the local time zone and the offset with the coordinate universal time shall be specified.

NOTE This construct is used to represent a moment of time whereas **time_measure** (see 21.3.22) represents amounts of time.

EXAMPLE 15:00 hours is an instant in time whereas 15 hours is an amount of time.

EXPRESS specification

```
*)
ENTITY local_time;
  hour_component  : hour_in_day;
  minute_component : OPTIONAL minute_in_hour;
  second_component : OPTIONAL second_in_minute;
  zone            : coordinated_universal_time_offset;
WHERE
  WR1: valid_time (SELF);
END_ENTITY; -- local_time
(*
```

Attribute definitions

hour_component: the number of hours.

minute_component: the number of minutes.

second_component: the number of seconds.

zone: the offset of the local time zone to the coordinated universal time.

Formal propositions:

WR1: The entity shall define a valid time.

The **second_component** attribute shall only exist if the **minute_component** attribute exists.

16.4.12 ordinal_date

An **ordinal_date** is a type of date defined as a day of a year.

EXPRESS specification

```

*)
ENTITY ordinal_date
  SUBTYPE OF (date);
  day_component : day_in_year_number;
WHERE
  WR1: (NOT leap_year(SELF.year_component) AND { 1 <= day_component <= 365 })
  OR
  (leap_year(SELF.year_component) AND { 1 <= day_component <= 366 });
END_ENTITY; -- ordinal_date
( *

```

Attribute definitions

day_component: the day element of the date.

Formal propositions:

WR1: The **day_component** shall be between 1 and 365 if the **year_component** is not a leap year. Otherwise the **day_component** shall be between 1 and 366.

16.4.13 relative_event_occurrence

A **relative_event_occurrence** is a type of **event_occurrence** that takes place with an offset to another **event_occurrence**.

EXAMPLE 'Five days after start of production' is an example for a **relative_event_occurrence**, where 'five days' is an offset relative to an **event_occurrence** 'start of production'.

EXPRESS specification

```

*)
ENTITY relative_event_occurrence
  SUBTYPE OF (event_occurrence);
  base_event : event_occurrence;
  offset : time_measure_with_unit;
END_ENTITY; -- relative_event_occurrence
( *

```

Attribute definitions

base_event: the **relative_event_occurrence** is defined with respect to the **base_event**.

offset: the amount of time that passes between the occurrence of the **base_event** and the **relative_event_occurrence**. A negative offset specifies that the **relative_event_occurrence** occurs before the **base_event**.

16.4.14 time_interval

A **time_interval** is the identification of an intervening time.

EXAMPLE 'Strike duration', 'delay of production', and 'Christmas holidays' are examples for **time_intervals**.

EXPRESS specification

```
*)
ENTITY time_interval;
  id          : identifier;
  name        : label;
  description  : OPTIONAL text;
END_ENTITY; -- time_interval
(*
```

Attribute definitions

id: the **identifier** that distinguishes the **time_interval**.

NOTE The context in which **id** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **time_interval** is known.

description: the **text** that characterizes the **time_interval**. The value of the attribute need not be specified.

16.4.15 time_interval_relationship

A **time_interval_relationship** relates two **time_intervals** with a description of their relationship.

NOTE 1 The role of **time_interval_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 This entity, together with the **time_interval** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY time_interval_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_time_interval : time_interval;
  related_time_interval  : time_interval;
END_ENTITY; -- time_interval_relationship
(*
```

Attribute definitions

name: the **label** by which the **time_interval_relationship** is known.

description: the **text** that characterizes the **time_interval_relationship**. The value of the attribute need not be specified.

relating_time_interval: one of the instances of **time_interval** that is a part of the relationship.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_time_interval: the other instance of **time_interval**. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

16.4.16 time_interval_role

A **time_interval_role** defines a role for an **time_interval_assignment** and a description of that role.

EXPRESS specification

```
*)
ENTITY time_interval_role;
  name          : label;
  description   : OPTIONAL text;
END_ENTITY; -- time_interval_role
(*
```

Attribute definitions

name: the **label** by which the **time_interval_role** is known.

description: the **text** that characterizes the **time_interval_role**. The value of the attribute need not be specified.

NOTE The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

16.4.17 time_interval_with_bounds

A **time_interval_with_bounds** is a type of **time_interval**. The **time_interval_with_bounds** is bounded either on one side or both sides. If neither **secondary_bound** nor **duration** are specified, the time interval begins at the point in time identified by **primary_bound** and has no specified end point.

EXAMPLE 'From 01-01-1999 to 31-12-1999' is an example for a both side bounded **time_interval_with_bounds**, 'from 01-01-2000 on', and 'until start of production' are examples for a one side bounded **time_intervals_with_bounds**.

EXPRESS specification

```
*)
ENTITY time_interval_with_bounds
  SUBTYPE OF (time_interval);
  primary_bound   : OPTIONAL date_time_or_event_occurrence;
  secondary_bound : OPTIONAL date_time_or_event_occurrence;
  duration        : OPTIONAL time_measure_with_unit;
WHERE
  WR1: NOT (EXISTS (secondary_bound) AND EXISTS (duration));
  WR2: EXISTS(primary_bound) OR EXISTS(secondary_bound);
END_ENTITY; -- time_interval_with_bounds
(*
```

Attribute definitions

primary_bound: one of the bounds of the **time_interval_with_bounds**. The value of this attribute need not be specified.

secondary_bound: the other bound of the **time_interval_with_bounds**. The value of this attribute need not be specified.

duration: the **time_measure_with_unit** that specifies the length of a **time_interval_with_bounds**. If positive, the **time_interval_with_bounds** is a lower bounded **time_interval_with_bounds**, else an upper bounded **time_interval_with_bounds**. The value of this attribute need not be specified.

Formal propositions:

WR1: The **secondary_bound** and **duration** shall not be specified both for one occurrence of **time_interval_with_bounds**.

NOTE This is enforced to prohibit redundant or inconsistent data.

WR2: Either the **primary_bound** or the **secondary_bound** or both shall be specified.

Informal proposition:

IP1: If **primary_bound** and **secondary_bound** are both specified, the point in time identified by **primary_bound** shall occur before the point in time identified by **secondary_bound**.

16.4.18 time_role

A **time_role** defines a role for a **time_assignment** and a description of that role.

EXPRESS specification

```
* )
ENTITY time_role;
  name          : label;
DERIVE
  description   : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                      'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- time_role
(*
```

Attribute definitions

name: the **label** by which the **time_role** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **time_role**.

NOTE 2 This attribute is an upwardly compatible addition to **time_role** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **time_role** shall be the described_item in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

16.4.19 week_of_year_and_day_date

A **week_of_year_and_date** is a **date** defined by identifying the week within a year and the day within that particular week.

EXPRESS specification

```
*)
ENTITY week_of_year_and_day_date
  SUBTYPE OF (date);
  week_component : week_in_year_number;
  day_component  : OPTIONAL day_in_week_number;
END_ENTITY; -- week_of_year_and_day_date
(*
```

Attribute definitions

week_component: the week element of the **date**.

day_component: the day element of the **date**.

Informal propositions:

valid_year_and_day: The result of the equation $\text{day_component} + (7 * \text{month_component})$ shall be between 1 and 365 if the year_component is not a leap year. Otherwise, the result of this equation shall be between 1 and 366.

16.5 Date time function definitions

16.5.1 acyclic_event_occurrence_relationship

The **acyclic_event_occurrence_relationship** function determines whether the graph of instances of the entity data type **event_occurrence** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **event_occurrence_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **event_occurrence_relationship** entity include rules that use this function.

EXPRESS specification

```
*)
FUNCTION acyclic_event_occurrence_relationship
  (relation      : event_occurrence_relationship;
   relatives    : SET [1:?] OF event_occurrence;
```

```

specific_relation : STRING) : BOOLEAN;

LOCAL
  x          : SET OF event_occurrence_relationship;
END_LOCAL;

IF relation.relating_event IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (evnt <* bag_to_set
            (USEDIN (relation.relating_event,
                    'DATE_TIME_SCHEMA.' +
                    'EVENT_OCCURRENCE_RELATIONSHIP.' +
                    'RELATED_EVENT')) |
            specific_relation IN TYPEOF (evnt));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_event_occurrence_relationship
    (x[i],
     relatives + relation.relating_event,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_event_occurrence_relationship
(*)

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **event_occurrence** for which the function is searching in the **relating_event_occurrence** parameter of the relation argument.

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

16.5.2 acyclic_time_interval_relationship

The **acyclic_time_interval_relationship** function determines whether the graph of instances of the entity data type **time_interval** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **time_interval_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **time_interval_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_time_interval_relationship
  (relation          : time_interval_relationship;
   relatives         : SET [1:?] OF time_interval;
   specific_relation : STRING) : BOOLEAN;

LOCAL  x          : SET OF time_interval_relationship;

```

```

END_LOCAL;

IF relation.relatering_time_interval IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (ti <* bag_to_set
            (USEDIN (relation.relatering_time_interval,
                    'DATE_TIME_SCHEMA.' +
                    'TIME_INTERVAL_RELATIONSHIP.' +
                    'RELATED_TIME_INTERVAL')) |
            specific_relation IN TYPEOF (ti));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_time_interval_relationship
    (x[i],
     relatives + relation.relatering_time_interval,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_time_interval_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **time_interval** for which the function is searching in the **relatering_time_interval** parameter of the relation argument.

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

16.5.3 leap_year

The **leap_year** function determines whether a given year is a leap year or not according to the Gregorian calendar algorithm. It returns TRUE if the year is a leap year. Otherwise, it returns FALSE.

EXPRESS specification

```

*)
FUNCTION leap_year (year : year_number) : BOOLEAN;
  IF (((year MOD 4) = 0) AND ((year MOD 100) <> 0)) OR
    ((year MOD 400) = 0) THEN
    RETURN(TRUE);
  ELSE
    RETURN(FALSE);
  END_IF;
END_FUNCTION; -- leap_year
(*

```

Argument definitions:

year: (input) the candidate **year_number** that is being checked.

16.5.4 valid_calendar_date

The **valid_calendar_date** function determines whether the components of a **calendar_date** indicate a valid **date**. If the **calendar_date** is valid, the function returns TRUE. Otherwise it returns FALSE.

EXPRESS specification

```

*)
FUNCTION valid_calendar_date (date : calendar_date) : LOGICAL;
  CASE date.month_component OF
    1 : RETURN( { 1 <= date.day_component <= 31 } );
    2 : BEGIN
        IF (leap_year(date.year_component)) THEN
          RETURN( { 1 <= date.day_component <= 29 } );
        ELSE
          RETURN( { 1 <= date.day_component <= 28 } );
        END_IF;
      END;
    3 : RETURN( { 1 <= date.day_component <= 31 } );
    4 : RETURN( { 1 <= date.day_component <= 30 } );
    5 : RETURN( { 1 <= date.day_component <= 31 } );
    6 : RETURN( { 1 <= date.day_component <= 30 } );
    7 : RETURN( { 1 <= date.day_component <= 31 } );
    8 : RETURN( { 1 <= date.day_component <= 31 } );
    9 : RETURN( { 1 <= date.day_component <= 30 } );
    10 : RETURN( { 1 <= date.day_component <= 31 } );
    11 : RETURN( { 1 <= date.day_component <= 30 } );
    12 : RETURN( { 1 <= date.day_component <= 31 } );
  END_CASE;
  RETURN (FALSE);
END_FUNCTION; -- valid_calendar_date
(*)

```

Argument definitions:

date: (input) the candidate **calendar_date** that is to be checked.

16.5.5 valid_time

The **valid_time** function determines whether a candidate **local_time** has a **minute_component** if it has a **second_component**. It returns **FALSE** if the condition is not met. Otherwise it returns **TRUE**.

EXPRESS specification

```

*)
FUNCTION valid_time (time: local_time) : BOOLEAN;
  IF EXISTS (time.second_component) THEN
    RETURN (EXISTS (time.minute_component));
  ELSE
    RETURN (TRUE);
  END_IF;
END_FUNCTION; -- valid_time
(*)

```

Argument definitions:

time: (input) the candidate **local_time** that is to be checked.

EXPRESS specification

```

*)
END_SCHEMA; -- date_time_schema
(*)

```

17 Group

The following EXPRESS declaration begins the **group_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA group_schema;

REFERENCE FROM support_resource_schema          -- ISO 10303-41
  (label,
   bag_to_set,
   identifier,
   text);
REFERENCE FROM basic_attribute_schema          -- ISO 10303-41
  (get_id_value,
   id_attribute);
(*)

```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

support_resource_schema clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema.

NOTE 3 This schema contains support resources.

17.1 Introduction

The subject of the **group_schema** is the definition of groups of items.

17.2 Fundamental concepts and assumptions

Groups are collections of product data that have a common identity. Any kinds of product data may be grouped together. The structure specified in this schema enables the common identity to be captured; it does not specify, or allow the specification of, the meaning of the common identity or the criteria for the common identity.

17.3 Group entity definitions

17.3.1 group

A **group** is an identification of a collection of elements.

EXPRESS specification

```

*)
ENTITY group;
  name      : label;
  description : OPTIONAL text;
DERIVE
  id          : identifier := get_id_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- group

```

(*

Attribute definitions

name: the **label** by which the **group** is known.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

description: the **text** that characterizes the **group**. The value of the attribute need not be specified.

id: the **identifier** that distinguishes the **group**. The value of this attribute need not be specified.

NOTE 2 This attribute is an upwardly compatible addition to **group** as specified in ISO 10303-41:1994.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 4 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

Formal propositions:

WR1: Each **group** shall be the identified_item in at most one **id_attribute**.

NOTE 5 The **id_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 6 A template for constraining the population of the entity data types defined in the **basic_attribute - schema** is described in Annex E.

17.3.2 **group_relationship**

A **group_relationship** relates two instances of the entity data type **group** with a description of their relationship.

NOTE 1 The role of **group_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **group** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY group_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_group : group;
  related_group  : group;
END_ENTITY; -- group_relationship
(*)
```

Attribute definitions

name: the **label** by which the **group_relationship** is known.

description: the **text** that characterizes the **group_relationship**. The value of the attribute need not be specified.

relating_group: one of the instances of **group** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_group: the other instance of **group** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

17.4 Group function definition: **acyclic_group_relationship**

The **acyclic_group_relationship** function determines whether the graph of instances of the entity data type **group** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **group_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **group_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_group_relationship
  (relation      : group_relationship;
   relatives     : SET [1:?] OF group;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x      : SET OF group_relationship;
END_LOCAL;

IF relation.relating_group IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (grp <* bag_to_set
            (USEDIN (relation.relating_group,
                    'GROUP_SCHEMA.' +
                    'GROUP_RELATIONSHIP.' +
                    'RELATED_GROUP')) |
            specific_relation IN TYPEOF (grp));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_group_relationship
    (x[i],
     relatives + relation.relating_group,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;

```

```

    END_REPEAT;
    RETURN(TRUE);
END_FUNCTION; -- acyclic_group_relationship
(*)

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **group** for which the function is searching in the relating_group parameter of the relation argument.

specific_relation: (input) the fully qualified name of a subtype of the **group_relationship** entity.

EXPRESS specification

```

*)
END_SCHEMA; -- group_schema
(*)

```

18 Effectivity

The following EXPRESS declaration begins the **effectivity_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA effectivity_schema;

REFERENCE FROM basic_attribute_schema -- ISO 10303-41
(description_attribute,
 get_description_value,
 get_name_value,
 name_attribute);

REFERENCE FROM date_time_schema -- ISO 10303-41
(date_time_or_event_occurrence,
 time_interval);

REFERENCE FROM measure_schema -- ISO 10303-41
(measure_with_unit);

REFERENCE FROM support_resource_schema -- ISO 10303-41
(bag_to_set,
 identifier,
 label,
 text);
(*)

```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
date_time_schema	clause 16 of this part of ISO 10303
measure_schema	clause 21 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

18.1 Introduction

The subject of the **effectivity_schema** is the validity of aspects of product data or product life cycle activity data specified against certain criteria.

This schema supports the representation of effectivity according to the following criteria:

- selection of a given batch of product data;
- selection of product data by serial number;
- selection of product data by a date.

18.2 Fundamental concepts and assumptions

Effectivity information can be attached to any aspect of product data or product life cycle activity data.

Only effectivities based on date, serial number, and lot number are considered in this schema.

18.3 Effectivity entity definitions

18.3.1 dated_effectivity

A **dated_effectivity** is a type of **effectivity** for which the domain of applicability is defined as a possibly open-ended interval of time.

EXPRESS specification

```
*)
ENTITY dated_effectivity
  SUBTYPE OF (effectivity);
  effectivity_end_date : OPTIONAL date_time_or_event_occurrence;
  effectivity_start_date : date_time_or_event_occurrence;
END_ENTITY; -- dated_effectivity
(*
```

Attribute definitions

effectivity_start_date: the **date_and_time** that defines the lower bound of the interval of applicability.

effectivity_end_date: the **date_and_time** that defines the upper bound of the interval of applicability. If a value for this attribute is not defined, the interval of applicability has no upper limit.

18.3.2 effectivity

An **effectivity** is the identification of a domain of applicability for product data.

EXAMPLE A process is effective from 1998 March 1 through 1998 December 31.

EXPRESS specification

```

*)
ENTITY effectivity
  SUPERTYPE OF (ONEOF (serial_numbered_effectivity,
                        dated_effectivity,
                        lot_effectivity,
                        time_interval_based_effectivity));
  id          : identifier;
DERIVE
  name        : label := get_name_value (SELF);
  description : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                        'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                        'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- effectivity
(*)

```

Attribute definitions

id: the **identifier** that distinguishes the **effectivity**.

NOTE 1 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

NOTE 2 The context in which id is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

name: the **label** by which the **effectivity** is known.

NOTE 3 This attribute is an upwardly compatible addition to **effectivity** as specified in ISO 10303-41:1994.

description: the **text** that characterizes the **effectivity**.

NOTE 4 This attribute is an upwardly compatible addition to **effectivity** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **effectivity** shall be the **described_item** in at most one **description_attribute**.

NOTE 5 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

WR2: Each **effectivity** shall be the **named_item** in at most one **name_attribute**.

NOTE 6 The **name_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 7 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

18.3.3 effectivity_relationship

An **effectivity_relationship** relates two instances of the entity data type **effectivity** with a description of their relationship.

NOTE 1 The role of **effectivity_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **effectivity** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY effectivity_relationship;
  name          : label;
  description    : OPTIONAL text;
  related_effectivity : effectivity;
  relating_effectivity : effectivity;
END_ENTITY; -- effectivity_relationship
(*
```

Attribute definitions

name: the **label** by which the **effectivity_relationship** is known.

description: the **text** that characterizes the **effectivity_relationship**. The value of the attribute need not be specified.

relating_effectivity: one of the instances of **effectivity** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_effectivity: the other instance of **effectivity** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

18.3.4 lot_effectivity

A **lot_effectivity** is a type of **effectivity** for which the domain of applicability is defined as a given batch of items.

EXPRESS specification

```
*)
ENTITY lot_effectivity
  SUBTYPE OF (effectivity);
  effectivity_lot_id      : identifier;
  effectivity_lot_size    : measure_with_unit;
END_ENTITY; -- lot_effectivity
```

(*

Attribute definitions

effectivity_lot_id: the identification of the batch of items.

effectivity_lot_size: the size of the batch of items.

18.3.5 serial_numbered_effectivity

A **serial_numbered_effectivity** is a type of **effectivity** for which the domain of applicability is defined as a possibly open-ended interval of serial numbers.

NOTE A serial number is represented in this International Standard as an identifier. It is supposed here that, regardless their format, identifiers used to represent serial numbers are ordered.

EXPRESS specification

```
*)
ENTITY serial_numbered_effectivity
  SUBTYPE OF (effectivity);
  effectivity_start_id : identifier;
  effectivity_end_id : OPTIONAL identifier;
END_ENTITY; -- serial_numbered_effectivity
(*
```

Attribute definitions

effectivity_start_id: the first valid serial number.

effectivity_end_id: the last valid serial number. If a value for this attribute is not defined, the interval of applicability has no upper bound.

18.3.6 time_interval_based_effectivity

A **time_interval_based_effectivity** is a type of effectivity for which the domain of applicability is defined as a **time_interval**.

NOTE This entity data type provides greater functionalities than the entity data type **dated_effectivity**. In particular, it allows the specification of an effectivity period in which the period of time is defined by a date and a duration.

EXPRESS specification

```
*)
ENTITY time_interval_based_effectivity
  SUBTYPE OF (effectivity);
  effectivity_period : time_interval;
END_ENTITY; -- time_interval_based_effectivity
(*
```

Attribute definitions

effectivity_period: the **time_interval** defining the period of time when the associated product data are effective.

18.4 Effectivity function definition: `acyclic_effectivity_relationship`

The `acyclic_effectivity_relationship` function determines whether the graph of instances of the entity data type `effectivity` that contains `relation` as one of its links contains a cycle. This function may be used to evaluate either a `effectivity_relationship` or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the `effectivity_relationship` entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_effectivity_relationship
  (relation      : effectivity_relationship;
   relatives     : SET [1:?] OF effectivity;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x      : SET OF effectivity_relationship;
END_LOCAL;

IF relation.relativity_effectivity IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (eff <* bag_to_set
            (USEDIN (relation.relativity_effectivity,
                    'EFFECTIVITY_SCHEMA.' +
                    'EFFECTIVITY_RELATIONSHIP.' +
                    'RELATED_EFFECTIVITY.')) |
            specific_relation IN TYPEOF (eff));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_effectivity_relationship
    (x[i],
     relatives + relation.relativity_effectivity,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_effectivity_relationship
(*)

```

Argument definitions:

relation: (input) the candidate `effectivity_relationship` to be checked.

relatives: (input) the set of instances of the entity data type `effectivity` for which the function is searching in the `relativity_effectivity` parameter of the relation argument.

specific_relation: (input) the fully qualified entity name of a type of `effectivity_relationship` entity.

EXPRESS specification

```
*)  
END_SCHEMA; -- effectivity_schema  
(*
```

19 External reference

The following EXPRESS declaration begins the **external_reference_schema** and identifies the necessary external references.

EXPRESS specification

```
*)  
SCHEMA external_reference_schema;  
  
REFERENCE FROM basic_attribute_schema          -- ISO 10303-41  
  (description_attribute,  
   get_description_value);  
  
REFERENCE FROM support_resource_schema         -- ISO 10303-41  
  (label,  
   text,  
   identifier,  
   bag_to_set);  
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

19.1 Introduction

The subject of the **external_reference_schema** is the means of identifying information that is not explicitly represented in a given exchange.

19.2 Fundamental concepts and assumptions

Product data may refer to information that is not explicitly represented in a given exchange. This information is either predefined, in the annotated EXPRESS schema to which the exchange conforms, or is defined elsewhere. If the information is defined in the annotated EXPRESS schema to which the exchange conforms, this requirement is satisfied by a reference that identifies the relevant information in the annotated EXPRESS schema. Otherwise, this requirement is satisfied by a reference that identifies the relevant information and its source.

19.3 External reference type definitions

19.3.1 message

A **message** is a communication that is addressed to a system in order to trigger some action. The result of such an action is an external item or identification.

NOTE The role of **message** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXPRESS specification

```
*)
TYPE message = STRING;
END_TYPE; -- message
(*
```

19.3.2 source_item

A **source_item** type allows for the designation of an **identifier** or a **message**.

EXPRESS specification

```
*)
TYPE source_item = SELECT
  (identifier,
   message);
END_TYPE; -- source_item
(*
```

19.4 External reference entity definitions

19.4.1 external_source

An **external_source** is the identification of a source of product data that is not the EXPRESS schema to which the exchange conforms.

NOTE 1 The product data may conform to some other part of this International Standard.

EXPRESS specification

```
*)
ENTITY external_source;
  source_id : source_item;
DERIVE
  description : text := get_description_value (SELF);
WHERE
  WR1 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
    'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- external_source
(*
```

Attribute definitions

source_id: the identification of the **external_source**.

description: the **text** that characterizes the **external_source**.

NOTE 2 This attribute is an upwardly compatible addition to **external_source** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: Each **external_source** shall be the described_item in at most one **description_attribute**.

NOTE 3 The **description_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 4 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

19.4.2 external_source_relationship

An **external_source_relationship** relates two instances of the entity data type **external_source** with a description of their relationship.

NOTE 1 The role of **external_source_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

EXAMPLE One **external_source** may be a subset of another **external_source**. This entity may be specialized to specify such a relationship.

NOTE 2 This entity, together with the **external_source** entity, is based on the relationship template that is described in annex E.3

EXPRESS specification

```
* )
ENTITY external_source_relationship;
  name          : label;
  description   : OPTIONAL text;
  relating_source : external_source;
  related_source  : external_source;
END_ENTITY; -- external_source_relationship
(*
```

Attribute definitions

name: the **label** by which the **external_source_relationship** is known.

description: the **text** that characterizes the **external_source_relationship**. The value of the attribute need not be specified.

relating_source: one of the instances of **external_source** that is a part of the relationship.

NOTE 3 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_source: the other instance of **external_source** that is a part of the relationship. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

19.4.3 externally_defined_item

An **externally_defined_item** is the identification of information that is not explicitly represented in a given exchange and that is not defined in the annotated EXPRESS schema to which the exchange conforms.

EXPRESS specification

```
*)
ENTITY externally_defined_item;
  item_id : source_item;
  source  : external_source;
END_ENTITY; -- externally_defined_item
(*
```

Attribute definitions

item_id: the string that specifies the identification of the referent item.

source: an **external_source** that contains the referent item.

19.4.4 externally_defined_item_relationship

An **externally_defined_item_relationship** relates two instances of the entity data type **externally_defined_item** with an identification and description of their relationship.

NOTE 1 The role of **externally_defined_item_relationship** can be defined in the annotated EXPRESS schemas that use or specialize this entity, or by default, in an agreement of common understanding between the partners sharing this information.

NOTE 2 Relationships represented using this entity may be parent child relationship. Annotated EXPRESS schemas that use or specialize this entity specify whether or not the relationship is directed.

NOTE 3 This entity, together with the **externally_defined_item** entity, is based on the relationship template that is described in annex E.3.

EXPRESS specification

```
*)
ENTITY externally_defined_item_relationship;
  name           : label;
  description    : OPTIONAL text;
  relating_item  : externally_defined_item;
  related_item   : externally_defined_item;
END_ENTITY; -- externally_defined_item_relationship
(*
```

Attribute definitions

name: the **label** by which the **externally_defined_item_relationship** is known.

description: the **text** that characterizes the **externally_defined_item_relationship**. The value of the attribute need not be specified.

relating_item: one of the instance of **externally_defined_item** that is a part of the relationship.

NOTE 4 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

related_item: the other instance of **externally_defined_item** that is a part of the relationship. If one element of the relationship is dependent up on the other, this attribute shall be the dependent one.

NOTE 5 The meaning of this attribute can be defined in the annotated EXPRESS schemas that use or specialize this entity, or in an agreement of common understanding between the partners sharing this information.

19.4.5 pre_defined_item

A **pre_defined_item** is the identification of information that is not explicitly represented in a given exchange but that is defined in the annotated EXPRESS schema to which the exchange conforms.

EXAMPLE A reference to the colour 'red' without any definition of the associated red-green-blue values would be a **pre_defined_item** if the red-green-blue values of the colour 'red' were specified in the relevant annotated EXPRESS schema.

EXPRESS specification

```
* )
ENTITY pre_defined_item;
  name : label;
END_ENTITY; -- pre_defined_item
(*
```

Attribute definitions

name: the **label** by which the **pre_defined_item** is known.

19.5 External reference function definitions

19.5.1 acyclic_external_source_relationship

The **acyclic_external_source_relationship** function determines whether the graph of instances of the entity data type **external_source** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **external_source_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **external_source_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_external_source_relationship
  (relation          : external_source_relationship;
   relatives         : SET [1:?] OF external_source;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x          : SET OF external_source_relationship;
END_LOCAL;

IF relation.relying_source IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (es <* bag_to_set
            (USEDIN (relation.relying_source,
                    'EXTERNAL_REFERENCE_SCHEMA.' +
                    'EXTERNAL_SOURCE_RELATIONSHIP.' +
                    'RELATED_SOURCE'))) |
            specific_relation IN TYPEOF (es));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_external_source_relationship
    (x[i],
     relatives + relation.relying_source,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_external_source_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **external_source** for which the function is searching in the relating_external_source parameter of the relation argument.

specific_relation: (input) the fully qualified name of a subtype of the **external_source_relationship** entity.

19.5.2 **acyclic_externally_defined_item_relationship**

The **acyclic_externally_defined_item_relationship** function determines whether the graph of instances of the entity data type **externally_defined_item** that contains **relation** as one of its links contains a cycle. This function may be used to evaluate either a **externally_defined_item_relationship** or any of its subtypes.

The function returns TRUE if no cycle has been detected. Otherwise it returns FALSE.

NOTE 1 The algorithm of the function is explained in annex E.2.

NOTE 2 This function is not used in this schema. It is defined here because other ISO 10303 integrated resources and application protocols that use the **externally_defined_item_relationship** entity include rules that use this function.

EXPRESS specification

```

*)
FUNCTION acyclic_externally_defined_item_relationship
  (relation      : externally_defined_item_relationship;
   relatives     : SET [1:?] OF externally_defined_item;
   specific_relation : STRING) : BOOLEAN;

LOCAL
  x      : SET OF externally_defined_item_relationship;
END_LOCAL;

IF relation.relying_item IN relatives THEN
  RETURN (FALSE);
END_IF;
x := QUERY (edi <* bag_to_set
            (USEDIN (relation.relying_item,
                    'EXTERNAL_REFERENCE_SCHEMA.' +
                    'EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.' +
                    'RELATED_ITEM')) |
            specific_relation IN TYPEOF (edi));
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_externally_defined_item_relationship
    (x[i],
     relatives + relation.relying_item,
     specific_relation) THEN
    RETURN(FALSE);
  END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; -- acyclic_externally_defined_item_relationship
(*

```

Argument definitions:

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

relatives: (input) the set of instances of the entity data type **externally_defined_item** for which the function is searching in the **relying_externally_defined_item** parameter of the relation argument.

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

EXPRESS specification

```

*)
END_SCHEMA; -- external_reference_schema
(*

```

20 Support resource

The following EXPRESS declaration begins the **support_resource_schema**.

EXPRESS specification

```

*)
SCHEMA support_resource_schema;
(*

```

NOTE 1 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 2 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 3 This schema contains support resources.

20.1 Introduction

The subject of the **support_resource_schema** is a description of EXPRESS declarations that are shared by more than one of the ISO 10303 integrated resource schemas.

NOTE In contrast to elements of the generic product description resource schemas, which are also potentially referenced by more than one ISO 10303 integrated resource schema, the elements of this schema cannot exist without (that is, are existent-dependent upon) the entities that reference them.

20.2 Fundamental concepts and assumptions

An identifier is a means that enables to distinguish an information from other information of the same kind. A label is a text information assigned to an information in order to facilitate its access by a human being.

Product data may have an identification. The formats used to define identifiers are quite various. This International Standard defines identifiers as strings. It does not specify the internal structure and content of these strings.

Product data may have a label. The formats used to define labels are quite various. This International Standard defines identifiers as strings. It does not specify the internal structure and content of these strings.

20.3 Support resource type definitions

20.3.1 identifier

An **identifier** is a string suitable for identifying some product data.

NOTE An identifier may or may not have a natural-language meaning.

EXAMPLE In the context of a product, a part number or serial number would be an example of an identifier.

EXPRESS specification

```
*)
TYPE identifier = STRING;
END_TYPE; -- identifier
(*
```

20.3.2 label

A **label** is an alphanumeric string that represents the human-interpretable name of something and shall have a natural-language meaning.

EXAMPLE 'Smith', 'Widget Inc.', and 'Materials Test Laboratory' are examples of labels.

EXPRESS specification

```
*)
TYPE label = STRING;
END_TYPE; -- label
(*
```

20.3.3 text

A **text** is an alphanumeric string intended to be read and understood by a human being. It is for information purposes only.

EXPRESS specification

```
*)
TYPE text = STRING;
END_TYPE; -- text
(*
```

20.4 Support resource function definitions

20.4.1 bag_to_set

This function converts BAGs into SETs.

EXAMPLE This function can be used to convert the BAGs returned by the USEDIN function into SETs.

EXPRESS specification

```
*)
FUNCTION bag_to_set
  (the_bag : BAG OF GENERIC : intype) : SET OF GENERIC : intype;
  LOCAL
    the_set: SET OF GENERIC : intype := [];
  END_LOCAL;
  IF SIZEOF (the_bag) > 0 THEN
    REPEAT i := 1 to HIINDEX (the_bag);
      the_set := the_set + the_bag [i];
    END_REPEAT;
  END_IF;
  RETURN (the_set);
END_FUNCTION; -- bag_to_set
(*
```

Argument definitions:

the_bag: (input) the BAG that is to be converted into a SET.

20.4.2 type_check_function

This function compares the names of the data types of the instance that is referred to by the parameter **the_type** with the set of entity names specified by **sub_names**.

Depending on the value of **criterion**, it returns TRUE if:

— case 0: the intersection of the two aggregates is not empty.

NOTE 1 If **sub_names** contains the entity names of all the subtypes of a particular entity data type, this case can be used to ensure that no instance of the supertype exists.

— case 1: the intersection of the two aggregates is empty.

NOTE 2 If **sub_names** contains the entity names of subtypes of a particular entity data type, this function can be used to ensure that no instance of these subtypes exists.

— case 2: the intersection of the two aggregates contains exactly one name.

NOTE 3 If **sub_names** contains the entity names of all the subtypes of a particular entity data type, this function can be used to ensure that no complex entity instance of subtypes of the entity exists and that only instances of subtypes of the entity exist.

NOTE 4 The concept of complex entity instance is defined in ISO 10303-11.

— case 3: the intersection of the two aggregates contains at most one name.

NOTE 5 If **sub_names** contains the entity names of subtypes of a particular entity data type, this function can be used to ensure that no complex entity instance of these subtypes of the entity exists.

EXPRESS specification

```

*)
FUNCTION type_check_function
  ( the_type : GENERIC;
    sub_names: SET OF STRING;
    criterion: INTEGER ): LOGICAL;

--first, check the correctness of function arguments
IF (( NOT EXISTS ( the_type ) ) OR (NOT ({0<= criterion <=3})) OR
    (SIZEOF ( sub_names ) = 0 ) ) THEN RETURN (UNKNOWN);
ELSE
  CASE criterion OF
    0: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) > 0);
    1: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) = 0);
    2: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) = 1);
    3: RETURN (SIZEOF ( sub_names * TYPEOF (the_type) ) <= 1);
  END_CASE;
END_IF;

END_FUNCTION;
( *

```

Argument definitions:

the_type: (input) the instance whose data type is checked.

sub_names: (input) the set of names of the entity data types that are candidates for the type of **the_type**.

EXPRESS specification

```

*)
END_SCHEMA; -- support_resource_schema
( *

```

21 Measure

The following EXPRESS declaration begins the **measure_schema** and identifies the necessary external references.

EXPRESS specification

```
* )
SCHEMA measure_schema;

REFERENCE FROM basic_attribute_schema          -- ISO 10303-41
  (name_attribute,
   get_name_value);

REFERENCE FROM representation_schema           -- ISO 10303-43
  (representation_context);

REFERENCE FROM support_resource_schema;        -- ISO 10303-41
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

basic_attribute_schema	clause 22 of this part of ISO 10303
representation_schema	ISO 10303-43
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

21.1 Introduction

The subject of the **measure_schema** is the description of physical quantities.

EXAMPLE Density, length, force, and time are different kinds of physical quantities.

The resource constructs defined in this schema are based upon material in ISO 31 and ISO 1000.

21.2 Fundamental concepts and assumptions

The following requirements are supported by this schema:

- it shall be possible to specify the predefined kinds of physical quantity if the kind of a physical quantity is known at the time when the schema is being specified;
- it shall be possible to specify non-predefined kinds of physical quantity if the kind of a physical quantity is not known at the time when the schema is being specified.

EXAMPLE The elements of a list of material properties are physical quantities. Each element may be a different kind of physical quantity. The kind of each element is only decided upon at instantiation time.

21.3 Measure type definitions

21.3.1 amount_of_substance_measure

An **amount_of_substance_measure** is the value for the quantity of a substance when compared with the number of atoms in 0.012 kilogram of carbon 12.

EXPRESS specification

```
*)
TYPE amount_of_substance_measure = REAL;
END_TYPE; -- amount_of_substance_measure
(*
```

21.3.2 area_measure

An **area_measure** is the value of the extent of a surface.

EXPRESS specification

```
*)
TYPE area_measure = REAL;
END_TYPE; -- area_measure
(*
```

21.3.3 celsius_temperature_measure

A **celsius_temperature_measure** is the value for the degree of heat of a body.

EXPRESS specification

```
*)
TYPE celsius_temperature_measure = REAL;
END_TYPE; -- celsius_temperature_measure
(*
```

21.3.4 context_dependent_measure

A **context_dependent_measure** is the value of a physical quantity that may be interpreted based on the context in which it is used.

EXPRESS specification

```
*)
TYPE context_dependent_measure = REAL;
END_TYPE; -- context_dependent_measure
(*
```

21.3.5 count_measure

A **count_measure** is the value of a count.

EXPRESS specification

```
*)  
TYPE count_measure = NUMBER;  
END_TYPE; -- count_measure  
(*
```

21.3.6 **descriptive_measure**

A **descriptive_measure** is a textual value of a physical quantity.

EXPRESS specification

```
*)  
TYPE descriptive_measure= STRING;  
END_TYPE; -- descriptive_measure  
(*
```

21.3.7 **electric_current_measure**

An **electric_current_measure** is the value for the movement of electrically charged particles.

EXPRESS specification

```
*)  
TYPE electric_current_measure= REAL;  
END_TYPE; -- electric_current_measure  
(*
```

21.3.8 **length_measure**

A **length_measure** is the value of a distance.

EXPRESS specification

```
*)  
TYPE length_measure = REAL;  
END_TYPE; -- length_measure  
(*
```

21.3.9 **luminous_intensity_measure**

A **luminous_intensity_measure** is the value for the brightness of a body.

EXPRESS specification

```
*)  
TYPE luminous_intensity_measure = REAL;  
END_TYPE; -- luminous_intensity_measure  
(*
```

21.3.10 **mass_measure**

A **mass_measure** is the value of the amount of matter that a body contains.

EXPRESS specification

```

*)
TYPE mass_measure = REAL;
END_TYPE; -- mass_measure
( *

```

21.3.11 measure_value

A **measure_value** is a value as defined in ISO 31-0.

EXPRESS specification

```

*)
TYPE measure_value = SELECT
  (length_measure,
   mass_measure,
   time_measure,
   electric_current_measure,
   thermodynamic_temperature_measure,
   celsius_temperature_measure,
   amount_of_substance_measure,
   luminous_intensity_measure,
   plane_angle_measure,
   solid_angle_measure,
   area_measure,
   volume_measure,
   ratio_measure,
   parameter_value,
   numeric_measure,
   context_dependent_measure,
   descriptive_measure,
   positive_length_measure,
   positive_plane_angle_measure,
   positive_ratio_measure,
   count_measure);
END_TYPE; -- measure_value
( *

```

21.3.12 numeric_measure

A **numeric_measure** is the numeric value of a physical quantity.

EXPRESS specification

```

*)
TYPE numeric_measure = NUMBER;
END_TYPE; -- numeric_measure
( *

```

21.3.13 parameter_value

A **parameter_value** is the value that specifies the amount of a parameter in a parameter space.

EXPRESS specification

```

*)
TYPE parameter_value = REAL;
END_TYPE; -- parameter_value
( *

```

21.3.14 plane_angle_measure

A **plane_angle_measure** is the value of an angle in a plane.

EXPRESS specification

```
*)
TYPE plane_angle_measure = REAL;
END_TYPE; -- plane_angle_measure
(*
```

21.3.15 positive_length_measure

A **positive_length_measure** is a **length_measure** that is greater than zero.

EXPRESS specification

```
*)
TYPE positive_length_measure = length_measure;
WHERE
  WR1: SELF > 0.0;
END_TYPE; -- positive_length_measure
(*
```

Formal propositions:

WR1: The value shall be positive.

21.3.16 positive_plane_angle_measure

A **positive_plane_angle_measure** is a **plane_angle_measure** that is greater than zero.

EXPRESS specification

```
*)
TYPE positive_plane_angle_measure = plane_angle_measure;
WHERE
  WR1: SELF > 0.0;
END_TYPE; -- positive_plane_angle_measure
(*
```

Formal propositions:

WR1: The value shall be positive.

21.3.17 positive_ratio_measure

A **positive_ratio_measure** is a **ratio_measure** that is greater than zero.

EXPRESS specification

```
*)
TYPE positive_ratio_measure = ratio_measure;
WHERE
  WR1: SELF > 0.0;
END_TYPE; -- positive_ratio_measure
(*
```

Formal propositions:

WR1: The value shall be positive.

21.3.18 ratio_measure

A **ratio_measure** is the value of the relation between two physical quantities that are of the same kind.

EXPRESS specification

```
*)
TYPE ratio_measure = REAL;
END_TYPE; -- ratio_measure
(*
```

21.3.19 si_prefix

An **si_prefix** is the name of a prefix that may be associated with an **si_unit**. The definitions of SI prefixes are specified in ISO 1000.

EXPRESS specification

```
*)
TYPE si_prefix = ENUMERATION OF
  (exa,
   peta,
   tera,
   giga,
   mega,
   kilo,
   hecto,
   deca,
   deci,
   centi,
   milli,
   micro,
   nano,
   pico,
   femto,
   atto);
END_TYPE; -- si_prefix
(*
```

Enumerated item definitions:

exa: see ISO 1000.

peta: see ISO 1000.

tera: see ISO 1000.

giga: see ISO 1000.

mega: see ISO 1000.

kilo: see ISO 1000.

hecto: see ISO 1000.

deca: see ISO 1000.

deci: see ISO 1000.

centi: see ISO 1000.

milli: see ISO 1000.

micro: see ISO 1000.

nano: see ISO 1000.

pico: see ISO 1000.

femto: see ISO 1000.

atto: see ISO 1000.

21.3.20 si_unit_name

An **si_unit_name** is the name of an SI unit.

NOTE 1 The definitions of the names of SI units are specified in ISO 1000.

EXPRESS specification

*)

TYPE si_unit_name = ENUMERATION OF

(metre,
gram,
second,
ampere,
kelvin,
mole,
candela,
radian,
steradian,
hertz,
newton,
pascal,
joule,
watt,
coulomb,
volt,
farad,
ohm,
siemens,
weber,
tesla,
henry,
degree_Celsius,
lumen,
lux,
becquerel,
gray,
sievert);

END_TYPE; -- si_unit_name

(*

Enumerated item definitions:

metre: see ISO 1000.

gram: see ISO 1000.

NOTE 2 ISO 1000 gives "kilogram" as the SI unit name. This part of ISO 10303 uses "gram" as the SI unit name.

second: see ISO 1000.

ampere: see ISO 1000.

kelvin: see ISO 1000.

mole: see ISO 1000.

candela: see ISO 1000.

radian: see ISO 1000.

steradian: see ISO 1000.

hertz: see ISO 1000.

newton: see ISO 1000.

pascal: see ISO 1000.

joule: see ISO 1000.

watt: see ISO 1000.

coulomb: see ISO 1000.

volt: see ISO 1000.

farad: see ISO 1000.

ohm: see ISO 1000.

siemens: see ISO 1000.

weber: see ISO 1000.

tesla: see ISO 1000.

henry: see ISO 1000.

degree_Celsius: see ISO 1000.

lumen: see ISO 1000.

lux: see ISO 1000.

becquerel: see ISO 1000.

gray: see ISO 1000.

sievert: see ISO 1000.

21.3.21 **solid_angle_measure**

A **solid_angle_measure** is the value of a solid angle.

EXPRESS specification

```
*)
TYPE solid_angle_measure = REAL;
END_TYPE; -- solid_angle_measure
(*
```

21.3.22 **thermodynamic_temperature_measure**

A **thermodynamic_temperature_measure** is the value for the degree of heat of a body.

EXPRESS specification

```
*)
TYPE thermodynamic_temperature_measure = REAL;
END_TYPE; -- thermodynamic_temperature_measure
(*
```

21.3.23 **time_measure**

A **time_measure** is the value of the duration of a period.

EXPRESS specification

```
*)
TYPE time_measure = REAL;
END_TYPE; -- time_measure
(*
```

21.3.24 **unit**

A **unit** is a physical quantity, with a value of one.

NOTE A **unit** is used to express other physical quantities of the same kind.

EXPRESS specification

```
*)
TYPE unit = SELECT
  (named_unit,
   derived_unit);
END_TYPE; -- unit
(*
```

21.3.25 volume_measure

A **volume_measure** is the value of the solid content of a body.

EXPRESS specification

```
*)
TYPE volume_measure = REAL;
END_TYPE; -- volume_measure
(*
```

21.4 Measure entity definitions

21.4.1 amount_of_substance_measure_with_unit

An **amount_of_substance_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is an amount_of_substance as defined in ISO 31. This entity data type shall only be used to characterize a number of particles.

EXPRESS specification

```
*)
ENTITY amount_of_substance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.AMOUNT_OF_SUBSTANCE_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY; -- amount_of_substance_measure_with_unit
(*
```

Formal propositions:

WR1: The unit shall be an **amount_of_substance_unit**.

21.4.2 amount_of_substance_unit

An **amount_of_substance_unit** is a type of **named_unit** in which the number of elementary entities of a substance as compared to the number of atoms in 0.012 kilograms of carbon-12 is expressed..

NOTE 1 This definition applies to the SI quantity 'mole'. When the mole is used, the elementary entities, whose quantity is expressed, must be specified. They may be atoms, molecules, ions, electrons or other particles or specified groups of such particles (see ISO 31-8).

NOTE 2 This unit corresponds to one of the seven fundamental quantities as specified in ISO 1000.

EXPRESS specification

```
*)
ENTITY amount_of_substance_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
    (SELF\named_unit.dimensions.time_exponent = 0.0) AND
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
    (SELF\named_unit.dimensions.amount_of_substance_exponent = 1.0) AND
```

ISO 10303-41:2000(E)

```
(SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);  
END_ENTITY; -- amount_of_substance_unit  
(*
```

Formal propositions:

WR1: The dimensional exponent of amount of substance shall be equal to one and all the other dimensional exponents shall be equal to zero.

21.4.3 area_measure_with_unit

An **area_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is an area as defined in ISO 31.

EXPRESS specification

```
*)  
ENTITY area_measure_with_unit  
  SUBTYPE OF (measure_with_unit);  
WHERE  
  WR1: 'MEASURE_SCHEMA.AREA_UNIT' IN TYPEOF  
    (SELF\nmeasure_with_unit.unit_component);  
END_ENTITY; -- area_measure_with_unit  
(*
```

Formal propositions:

WR1: The unit shall be an **area_unit**.

21.4.4 area_unit

An **area_unit** is a type of **named_unit** in which the extent of a surface is expressed.

NOTE Use of this entity data type is deprecated as it is not usable. See the example in Annex F.

EXPRESS specification

```
*)  
ENTITY area_unit  
  SUBTYPE OF (named_unit);  
WHERE  
  WR1: (SELF\named_unit.dimensions.length_exponent = 2.0) AND  
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND  
    (SELF\named_unit.dimensions.time_exponent = 0.0) AND  
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND  
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND  
    (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND  
    (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);  
END_ENTITY; -- area_unit  
(*
```

Formal propositions:

WR1: The dimensional exponent of length shall be equal to two and all the other dimensional exponents shall be equal to zero.

21.4.5 celsius_temperature_measure_with_unit

A **celsius_temperature_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a Celsius temperature as defined in ISO 31-4.

EXPRESS specification

```

*)
ENTITY celsius_temperature_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.THERMODYNAMIC_TEMPERATURE_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY; -- celsius_temperature_measure_with_unit
(*

```

Formal propositions:

WR1: The unit shall be a **thermodynamic_temperature_unit**.

21.4.6 context_dependent_unit

A **context_dependent_unit** is a type of **named_unit** that is not related to the system of units defined in this part of ISO 10303.

EXAMPLE The number of parts in an assembly is a physical quantity that may be measured in a unit called "parts". Such a unit cannot be related to an SI unit.

EXPRESS specification

```

*)
ENTITY context_dependent_unit
  SUBTYPE OF (named_unit);
  name : label;
END_ENTITY; -- context_dependent_unit
(*

```

Attribute definitions

name: the **label** by which the **context_dependent_unit** is known.

21.4.7 conversion_based_unit

A **conversion_based_unit** is a type of **named_unit** that defines a unit on the basis of a **measure_with_unit**.

NOTE The **value_component** attribute of the **measure_with_unit** defines the conversion factor.

EXAMPLE An inch is a **conversion_based_unit**. It is from the Imperial system, its name is "inch", and it can be related to the **si_unit**, millimetre, through a **measure_with_unit** whose value is 25.4 millimetre. A foot is also a **conversion_based_unit**. It is from the Imperial system, its name is "foot", and it can be related to an **si_unit**, millimetre, either directly or through the unit called "inch".

EXPRESS specification

```

*)
ENTITY conversion_based_unit
  SUBTYPE OF (named_unit);
  name          : label;
  conversion_factor : measure_with_unit;
END_ENTITY; -- conversion_based_unit
(*

```

Attribute definitions

name: the **label** by which the **conversion_based_unit** is known.

conversion_factor: the **measure_with_unit** that specifies the physical quantity from which the **conversion_based_unit** is derived.

21.4.8 derived_unit

A **derived_unit** is an expression of units.

EXAMPLE Newtons per square millimetre is a **derived_unit**.

EXPRESS specification

```

*)
ENTITY derived_unit;
  elements : SET [1:?] OF derived_unit_element;
DERIVE
  name      : label := get_name_value (SELF);
WHERE
  WR1 : (SIZEOF (elements) > 1) OR
        ((SIZEOF (elements) = 1) AND (elements[1].exponent <> 1.0));
  WR2 : SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                      'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- derived_unit
(*

```

Attribute definitions

elements: the **derived_unit_element**s and their exponents that define the **derived_unit**.

name: the **label** by which the **derived_unit** is known.

NOTE 1 This attribute is an upwardly compatible addition to **derived_unit** as specified in ISO 10303-41:1994.

Formal propositions:

WR1: There shall be either more than one member in the elements set or the value of the exponent of the single element of the elements set shall not be equal to one.

WR2: Each **derived_unit** shall be the **named_item** in at most one **name_attribute**.

NOTE 2 The **name_attribute** data type is defined in clause 22 of this part of ISO 10303.

NOTE 3 A template for constraining the population of the entity data types defined in the **basic_attribute_schema** is described in Annex E.

21.4.9 derived_unit_element

A **derived_unit_element** is the association of a **named_unit** with an exponent.

NOTE This entity is used in this part of ISO 10303 to represent an element of the dimensional expression of a **derived_unit**.

EXAMPLE Newtons per square millimetre is a derived unit. It has two elements, Newton whose exponent has a value of 1, and millimetre whose exponent is -2.

EXPRESS specification

```
*)
ENTITY derived_unit_element;
  unit      : named_unit;
  exponent  : REAL;
END_ENTITY; -- derived_unit_element
(*
```

Attribute definitions

unit: the **named_unit** that specifies the mathematical factor of the element.

exponent: the power that is applied to the unit attribute.

21.4.10 dimensional_exponents

The **dimensional_exponents** entity defines the powers of the dimensions of the seven base quantities.

NOTE 1 The seven base quantities are defined in ISO 31.

NOTE 2 This entity enables the dimensional expression of any physical quantity with respect to these seven base quantities.

NOTE 3 Length, mass, time, electric current, thermodynamic temperature, amount of substance, and luminous intensity are the seven base quantities.

EXAMPLE 1 A length of 2 millimetres has a length exponent of 1. The remaining exponents are equal to 0.

EXAMPLE 2 A velocity of 2 millimetres per second has a length exponent of 1 and a time exponent of -1. The remaining exponents are 0.

EXPRESS specification

```
*)
ENTITY dimensional_exponents;
  length_exponent      : REAL;
  mass_exponent        : REAL;
  time_exponent        : REAL;
  electric_current_exponent : REAL;
  thermodynamic_temperature_exponent : REAL;
  amount_of_substance_exponent : REAL;
  luminous_intensity_exponent : REAL;
END_ENTITY; -- dimensional_exponents
(*
```

Attribute definitions

length_exponent: the power of the length base quantity.

mass_exponent: the power of the mass base quantity.

time_exponent: the power of the time base quantity.

electric_current_exponent: the power of the electric current base quantity.

thermodynamic_temperature_exponent: the power of the thermodynamic temperature base quantity.

amount_of_substance_exponent: the power of the amount of substance base quantity.

luminous_intensity_exponent: the power of the luminous intensity base quantity.

21.4.11 electric_current_measure_with_unit

An **electric_current_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is an **electric_current** as defined in ISO 31.

EXPRESS specification

```
*)
ENTITY electric_current_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.ELECTRIC_CURRENT_UNIT' IN TYPEOF
      (SELF\measure_with_unit.unit_component);
END_ENTITY; -- electric_current_measure_with_unit
(*)
```

Formal propositions:

WR1: The unit shall be an **electric_current_unit**.

21.4.12 electric_current_unit

An **electric_current_unit** is a type of **named_unit** in which the movement of electrically charged particles is expressed.

NOTE This unit corresponds to one of the seven fundamental quantities as specified in ISO 1000.

EXPRESS specification

```
*)
ENTITY electric_current_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
      (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
      (SELF\named_unit.dimensions.time_exponent = 0.0) AND
      (SELF\named_unit.dimensions.electric_current_exponent = 1.0) AND
      (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
      (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
      (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- electric_current_unit
```

(*

Formal propositions:

WR1: The dimensional exponent of electric current shall be equal to one and all the other dimensional exponents shall be equal to zero.

21.4.13 global_unit_assigned_context

A **global_unit_assigned_context** is a type of **representation_context** in which the units apply to all **measure_values** of the applicable kind.

EXPRESS specification

```
*)
ENTITY global_unit_assigned_context
  SUBTYPE OF (representation_context);
  units : SET [1:?] OF unit;
END_ENTITY; -- global_unit_assigned_context
(*
```

Attribute definitions

units: the units that apply in the **representation_context**.

Informal propositions:

unique_units: each unit shall be a different kind of unit.

NOTE For an example of the use of this entity, see F.4.6.

21.4.14 length_measure_with_unit

A **length_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a length as defined in ISO 31.

EXPRESS specification

```
*)
ENTITY length_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.LENGTH_UNIT' IN TYPEOF
      (SELF\measure_with_unit.unit_component);
END_ENTITY; -- length_measure_with_unit
(*
```

Formal propositions:

WR1: The unit shall be a **length_unit**.

21.4.15 length_unit

A **length_unit** is a type of **named_unit** in which distances are expressed.

NOTE This unit corresponds to one of the seven fundamental quantities as specified in ISO 1000.

EXPRESS specification

```

*)
ENTITY length_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1:(SELF\named_unit.dimensions.length_exponent = 1.0) AND
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
    (SELF\named_unit.dimensions.time_exponent = 0.0) AND
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
    (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
    (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- length_unit
(*

```

Formal propositions:

WR1: The dimensional exponent of length shall be equal to one and all the other dimensional exponents shall be equal to zero.

21.4.16 luminous_intensity_measure_with_unit

A **luminous_intensity_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a luminous_intensity as defined in ISO 31.

EXPRESS specification

```

*)
ENTITY luminous_intensity_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.LUMINOUS_INTENSITY_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY; -- luminous_intensity_measure_with_unit
(*

```

Formal propositions:

WR1: The unit shall be a **luminous_intensity_unit**.

21.4.17 luminous_intensity_unit

A **luminous_intensity_unit** is a type of **named_unit** in which the brightness of a body is expressed.

NOTE This unit corresponds to one of the seven fundamental quantities as specified in ISO 1000.

EXPRESS specification

```

*)
ENTITY luminous_intensity_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1:(SELF\named_unit.dimensions.length_exponent = 0.0) AND
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
    (SELF\named_unit.dimensions.time_exponent = 0.0) AND
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
    (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND

```

```
(SELF\named_unit.dimensions.luminous_intensity_exponent = 1.0);
END_ENTITY; -- luminous_intensity_unit
(*
```

Formal propositions:

WR1: The dimensional exponent of luminous intensity shall be equal to one and all the other dimensional exponents shall be equal to zero.

21.4.18 mass_measure_with_unit

A **mass_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a mass as defined in ISO 31.

EXPRESS specification

```
*)
ENTITY mass_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.MASS_UNIT' IN TYPEOF
      (SELF\measure_with_unit.unit_component);
END_ENTITY; -- mass_measure_with_unit
(*
```

Formal propositions:

WR1: The unit shall be a **mass_unit**.

21.4.19 mass_unit

A **mass_unit** is a type of **named_unit** in which the amount of matter that a body contains is expressed.

NOTE This unit corresponds to one of the seven fundamental quantities as specified in ISO 1000.

EXPRESS specification

```
*)
ENTITY mass_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
      (SELF\named_unit.dimensions.mass_exponent = 1.0) AND
      (SELF\named_unit.dimensions.time_exponent = 0.0) AND
      (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
      (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
      (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
      (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- mass_unit
(*
```

Formal propositions:

WR1: The dimensional exponent of mass shall be equal to one and all the other dimensional exponents shall be equal to zero.

21.4.20 **measure_with_unit**

A **measure_with_unit** is the specification of a physical quantity as defined in ISO 31.

EXPRESS specification

```

*)
ENTITY measure_with_unit
  SUPERTYPE OF (ONEOF (length_measure_with_unit,
                        mass_measure_with_unit,
                        time_measure_with_unit,
                        electric_current_measure_with_unit,
                        thermodynamic_temperature_measure_with_unit,
                        celsius_temperature_measure_with_unit,
                        amount_of_substance_measure_with_unit,
                        luminous_intensity_measure_with_unit,
                        plane_angle_measure_with_unit,
                        solid_angle_measure_with_unit,
                        area_measure_with_unit,
                        volume_measure_with_unit,
                        ratio_measure_with_unit ));
  value_component : measure_value;
  unit_component  : unit;
WHERE
  WR1: valid_units (SELF);
END_ENTITY; -- measure_with_unit
(*

```

Attribute definitions

value_component: the value of the physical quantity if expressed with respect to the unit_component.

unit_component: the unit in which the physical quantity is expressed.

Formal propositions:

WR1: The unit_component shall be a valid unit for the kind of measure specified by the value_component.

21.4.21 **named_unit**

A **named_unit** is a unit quantity.

EXPRESS specification

```

*)
ENTITY named_unit
  SUPERTYPE OF (ONEOF (si_unit,
                        conversion_based_unit,
                        context_dependent_unit)
                ANDOR
                ONEOF (length_unit,
                        mass_unit,
                        time_unit,
                        electric_current_unit,
                        thermodynamic_temperature_unit,
                        amount_of_substance_unit,
                        luminous_intensity_unit,
                        plane_angle_unit,
                        solid_angle_unit,

```

```

        area_unit,
        volume_unit,
        ratio_unit ));
    dimensions : dimensional_exponents;
END_ENTITY; -- named_unit
(*)

```

Attribute definitions

dimensions: the exponents of the base properties by which the **named_unit** is defined.

21.4.22 plane_angle_measure_with_unit

A **plane_angle_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a plane_angle as defined in ISO 31.

EXPRESS specification

```

*)
ENTITY plane_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.PLANE_ANGLE_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY; -- plane_angle_measure_with_unit
(*)

```

Formal propositions:

WR1: The unit shall be a **plane_angle_unit**.

21.4.23 plane_angle_unit

A **plane_angle_unit** is a type of **named_unit** in which angles in planes are expressed.

EXPRESS specification

```

*)
ENTITY plane_angle_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
    (SELF\named_unit.dimensions.time_exponent = 0.0) AND
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
    (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
    (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- plane_angle_unit
(*)

```

Formal propositions:

WR1: All the dimensional exponents shall be equal to zero.

21.4.24 ratio_measure_with_unit

A **ratio_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a ratio as defined in ISO 31.

EXPRESS specification

```
*)
ENTITY ratio_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.RATIO_UNIT' IN TYPEOF
      (SELF\measure_with_unit.unit_component);
END_ENTITY; -- ratio_measure_with_unit
(*
```

Formal propositions:

WR1: The unit shall be a **ratio_unit**.

21.4.25 ratio_unit

A **ratio_unit** is a type of **named_unit** in which the ratio between two physical quantities of the same kind is expressed.

EXPRESS specification

```
*)
ENTITY ratio_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
      (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
      (SELF\named_unit.dimensions.time_exponent = 0.0) AND
      (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
      (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
      (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
      (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- ratio_unit
(*
```

Formal propositions:

WR1: All the dimensional exponents shall be equal to zero.

21.4.26 si_unit

An **si_unit** is a type of **named_unit** that defines a unit with respect to the system of units defined in this schema.

NOTE The system of units is based on the specifications of ISO 1000 but differs from them for the unit of mass.

EXPRESS specification

```
*)
ENTITY si_unit
  SUBTYPE OF (named_unit);
  prefix : OPTIONAL si_prefix;
```

```

    name      : si_unit_name;
DERIVE
    SELF\named_unit.dimensions : dimensional_exponents
                                := dimensions_for_si_unit (name);
END_ENTITY; -- si_unit
(*)

```

Attribute definitions

prefix: the **si_prefix** that specifies the ratio with the unit specified by name.

name: the **label** by which the **si_unit_name** is known.

21.4.27 solid_angle_measure_with_unit

A **solid_angle_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a solid_angle as defined in ISO 31.

EXPRESS specification

```

*)
ENTITY solid_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.SOLID_ANGLE_UNIT' IN TYPEOF
        (SELF\named_unit.unit_component);
END_ENTITY; -- solid_angle_measure_with_unit
(*)

```

Formal propositions:

WR1: The unit shall be a **solid_angle_unit**.

21.4.28 solid_angle_unit

A **solid_angle_unit** is a type of **named_unit** in which solid angles are expressed.

EXPRESS specification

```

*)
ENTITY solid_angle_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
        (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
        (SELF\named_unit.dimensions.time_exponent = 0.0) AND
        (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
        (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
        (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
        (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- solid_angle_unit
(*)

```

Formal propositions:

WR1: All the dimensional exponents shall be equal to zero.

21.4.29 thermodynamic_temperature_measure_with_unit

A **thermodynamic_temperature_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a thermodynamic_temperature as defined in ISO 31.

EXPRESS specification

```
*)
ENTITY thermodynamic_temperature_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.THERMODYNAMIC_TEMPERATURE_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY; -- thermodynamic_temperature_measure_with_unit
(*)
```

Formal propositions:

WR1: The unit shall be a **thermodynamic_temperature_unit**.

21.4.30 thermodynamic_temperature_unit

A **thermodynamic_temperature_unit** is a type of **named_unit** in which the degree of heat of a body is expressed.

NOTE This unit corresponds to one of the seven fundamental quantities as specified in ISO 1000.

EXPRESS specification

```
*)
ENTITY thermodynamic_temperature_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
    (SELF\named_unit.dimensions.time_exponent = 0.0) AND
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 1.0) AND
    (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
    (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- thermodynamic_temperature_unit
(*)
```

Formal propositions:

WR1: The dimensional exponent of thermodynamic temperature shall be equal to one and the other dimensional exponents shall be equal to zero.

21.4.31 time_measure_with_unit

A **time_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a time as defined in ISO 31.

EXPRESS specification

```

*)
ENTITY time_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.TIME_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY; -- time_measure_with_unit
(*

```

Formal propositions:

WR1: The unit shall be a **time_unit**.

21.4.32 time_unit

A **time_unit** is a type of **named_unit** in which the duration of periods is expressed.

NOTE This unit corresponds to one of the seven fundamental quantities as specified in ISO 1000.

EXPRESS specification

```

*)
ENTITY time_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: (SELF\named_unit.dimensions.length_exponent = 0.0) AND
    (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
    (SELF\named_unit.dimensions.time_exponent = 1.0) AND
    (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
    (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
    (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
    (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- time_unit
(*

```

Formal propositions:

WR1: The dimensional exponent of time shall be equal to one and all the other dimensional exponents shall be equal to zero.

21.4.33 volume_measure_with_unit

A **volume_measure_with_unit** is a type of **measure_with_unit** in which the physical quantity is a volume as defined in ISO 31.

EXPRESS specification

```

*)
ENTITY volume_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.VOLUME_UNIT' IN TYPEOF
    (SELF\measure_with_unit.unit_component);
END_ENTITY; -- volume_measure_with_unit
(*

```

Formal propositions:

WR1: The unit shall be a **volume_unit**.

21.4.34 volume_unit

A **volume_unit** is a type of **named_unit** in which the solid content of a body is expressed.

NOTE Use of this entity data type is deprecated as it is not usable. See the example in Annex F

EXPRESS specification

```

*)
ENTITY volume_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1:(SELF\named_unit.dimensions.length_exponent = 3.0) AND
  (SELF\named_unit.dimensions.mass_exponent = 0.0) AND
  (SELF\named_unit.dimensions.time_exponent = 0.0) AND
  (SELF\named_unit.dimensions.electric_current_exponent = 0.0) AND
  (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0.0) AND
  (SELF\named_unit.dimensions.amount_of_substance_exponent = 0.0) AND
  (SELF\named_unit.dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY; -- volume_unit
(*)

```

Formal propositions:

WR1: The dimensional exponent of length shall be equal to three and all the other dimensional exponents shall be equal to zero.

21.5 Measure function definitions

21.5.1 derive_dimensional_exponents

The **derive_dimensional_exponents** function determines the dimensional exponents of a unit. For named units the dimensions attribute is returned and for derived units the dimensional exponents are calculated from its elements.

EXPRESS specification

```

*)
FUNCTION derive_dimensional_exponents (x : unit) : dimensional_exponents;

LOCAL
  result : dimensional_exponents :=
    dimensional_exponents(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0);
END_LOCAL;

IF 'MEASURE_SCHEMA.DERIVED_UNIT' IN TYPEOF(x) THEN -- x is a derived unit
  REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements);

    result.length_exponent :=
      result.length_exponent +
      (x.elements[i].exponent *
      x.elements[i].unit.dimensions.length_exponent);

    result.mass_exponent :=
      result.mass_exponent +
      (x.elements[i].exponent *
      x.elements[i].unit.dimensions.mass_exponent);
  END REPEAT;
END FUNCTION;

```

```

result.time_exponent :=
  result.time_exponent +
  (x.elements[i].exponent *
  x.elements[i].unit.dimensions.time_exponent);

result.electric_current_exponent      :=
  result.electric_current_exponent +
  (x.elements[i].exponent *
  x.elements[i].unit.dimensions.electric_current_exponent);

result.thermodynamic_temperature_exponent :=
  result.thermodynamic_temperature_exponent +
  (x.elements[i].exponent *
  x.elements[i].unit.dimensions.thermodynamic_temperature_exponent);

result.amount_of_substance_exponent    :=
  result.amount_of_substance_exponent +
  (x.elements[i].exponent *
  x.elements[i].unit.dimensions.amount_of_substance_exponent);

result.luminous_intensity_exponent     :=
  result.luminous_intensity_exponent +
  (x.elements[i].exponent *
  x.elements[i].unit.dimensions.luminous_intensity_exponent);

  END_REPEAT;
ELSE -- x is a unitless or a named unit
  result := x.dimensions;
END_IF;
RETURN (result);
END_FUNCTION; -- derive_dimensional_exponents
(*)

```

Argument definitions:

x: (input) the unit that the **dimensional_exponents** are being derived from.

21.5.2 dimensions_for_si_unit

The **dimensions_for_si_unit** function returns the **dimensional_exponents** of the given **si_unit**.

EXPRESS specification

```

*)
FUNCTION dimensions_for_si_unit (n : si_unit_name) : dimensional_exponents;
  CASE n OF
    metre      : RETURN (dimensional_exponents
      (1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0));
    gram       : RETURN (dimensional_exponents
      (0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0));
    second     : RETURN (dimensional_exponents
      (0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0));
    ampere     : RETURN (dimensional_exponents
      (0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0));
    kelvin     : RETURN (dimensional_exponents
      (0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0));
    mole       : RETURN (dimensional_exponents
      (0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0));
    candela    : RETURN (dimensional_exponents
      (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0));
    radian     : RETURN (dimensional_exponents
      (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0));
    steradian  : RETURN (dimensional_exponents

```

```

                                (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0));
hertz      : RETURN (dimensional_exponents
                                (0.0, 0.0, -1.0, 0.0, 0.0, 0.0, 0.0));
newton     : RETURN (dimensional_exponents
                                (1.0, 1.0, -2.0, 0.0, 0.0, 0.0, 0.0));
pascal     : RETURN (dimensional_exponents
                                (-1.0, 1.0, -2.0, 0.0, 0.0, 0.0, 0.0));
joule      : RETURN (dimensional_exponents
                                (2.0, 1.0, -2.0, 0.0, 0.0, 0.0, 0.0));
watt       : RETURN (dimensional_exponents
                                (2.0, 1.0, -3.0, 0.0, 0.0, 0.0, 0.0));
coulomb    : RETURN (dimensional_exponents
                                (0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0));
volt       : RETURN (dimensional_exponents
                                (2.0, 1.0, -3.0, -1.0, 0.0, 0.0, 0.0));
farad      : RETURN (dimensional_exponents
                                (-2.0, -1.0, 4.0, 1.0, 0.0, 0.0, 0.0));
ohm        : RETURN (dimensional_exponents
                                (2.0, 1.0, -3.0, -2.0, 0.0, 0.0, 0.0));
siemens    : RETURN (dimensional_exponents
                                (-2.0, -1.0, 3.0, 2.0, 0.0, 0.0, 0.0));
weber      : RETURN (dimensional_exponents
                                (2.0, 1.0, -2.0, -1.0, 0.0, 0.0, 0.0));
tesla      : RETURN (dimensional_exponents
                                (0.0, 1.0, -2.0, -1.0, 0.0, 0.0, 0.0));
henry      : RETURN (dimensional_exponents
                                (2.0, 1.0, -2.0, -2.0, 0.0, 0.0, 0.0));
degree_Celsius : RETURN (dimensional_exponents
                                (0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0));
lumen      : RETURN (dimensional_exponents
                                (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0));
lux        : RETURN (dimensional_exponents
                                (-2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0));
becquerel  : RETURN (dimensional_exponents
                                (0.0, 0.0, -1.0, 0.0, 0.0, 0.0, 0.0));
gray       : RETURN (dimensional_exponents
                                (2.0, 0.0, -2.0, 0.0, 0.0, 0.0, 0.0));
sievert    : RETURN (dimensional_exponents
                                (2.0, 0.0, -2.0, 0.0, 0.0, 0.0, 0.0));
OTHERWISE  : RETURN (?);
END_CASE;
END_FUNCTION; -- dimensions_for_si_unit
(*)

```

Argument definitions:

n : (input) the name of the **unit** for which the **dimensional_exponents** will be returned.

21.5.3 valid_units

The **valid_units** function validates a **measure_with_unit**. If the unit of the **measure_with_unit** is valid the function returns TRUE. Otherwise, it returns FALSE.

EXPRESS specification

```

*)
FUNCTION valid_units ( m : measure_with_unit ) : BOOLEAN ;
  IF 'MEASURE_SCHEMA.LENGTH_MEASURE' IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
      dimensional_exponents ( 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
      RETURN ( FALSE );
    END_IF;
  END_IF;
END_FUNCTION;

```

```

IF 'MEASURE_SCHEMA.MASS_MEASURE' IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.TIME_MEASURE' IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.ELECTRIC_CURRENT_MEASURE'
IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.THERMODYNAMIC_TEMPERATURE_MEASURE'
IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.CELSIUS_TEMPERATURE_MEASURE'
IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.AMOUNT_OF_SUBSTANCE_MEASURE'
IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.LUMINOUS_INTENSITY_MEASURE'
IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.PLANE_ANGLE_MEASURE' IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.SOLID_ANGLE_MEASURE' IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.AREA_MEASURE' IN TYPEOF ( m.value_component ) THEN
  IF derive_dimensional_exponents ( m.unit_component ) <>
    dimensional_exponents ( 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
    RETURN ( FALSE );
  END_IF;
END_IF;
IF 'MEASURE_SCHEMA.VOLUME_MEASURE' IN TYPEOF ( m.value_component ) THEN

```

```

    IF derive_dimensional_exponents ( m.unit_component ) <>
        dimensional_exponents ( 3.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
        RETURN ( FALSE );
    END_IF;
END_IF;
IF 'MEASURE_SCHEMA.RATIO_MEASURE' IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
        dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
        RETURN ( FALSE );
    END_IF;
END_IF;
IF 'MEASURE_SCHEMA.POSITIVE_LENGTH_MEASURE'
IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
        dimensional_exponents ( 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
        RETURN ( FALSE );
    END_IF;
END_IF;
IF 'MEASURE_SCHEMA.POSITIVE_PLANE_ANGLE_MEASURE'
IN TYPEOF ( m.value_component ) THEN
    IF derive_dimensional_exponents ( m.unit_component ) <>
        dimensional_exponents ( 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ) THEN
        RETURN ( FALSE );
    END_IF;
END_IF;
RETURN ( TRUE );
END_FUNCTION; -- valid_units
( *

```

Argument definitions:

m: (input) the candidate **measure_with_unit** that is to be checked.

EXPRESS specification

```

*)
END_SCHEMA; -- measure_schema
( *

```

22 Basic attribute

The following EXPRESS declaration begins the **basic_attribute_schema** and identifies the necessary external references.

EXPRESS specification

```

*)
SCHEMA basic_attribute_schema;

REFERENCE FROM action_schema -- ISO 10303-41
    (action,
     action_request_solution);

REFERENCE FROM application_context_schema -- ISO 10303-41
    (application_context);

REFERENCE FROM approval_schema -- ISO 10303-41
    (approval_date_time,
     approval_role);

REFERENCE FROM configuration_management_schema -- ISO 10303-44

```

```

(configuration_design);

REFERENCE FROM date_time_schema -- ISO 10303-41
(date_role,
 date_time_role,
 time_role);

REFERENCE FROM effectivity_schema -- ISO 10303-41
(effectivity);

REFERENCE FROM external_reference_schema -- ISO 10303-41
(external_source);

REFERENCE FROM management_resources_schema -- ISO 10303-41
(action_assignment,
 action_request_assignment,
 approval_assignment,
 certification_assignment,
 contract_assignment,
 document_reference,
 effectivity_assignment,
 external_referent_assignment,
 group_assignment,
 name_assignment,
 security_classification_assignment);

REFERENCE FROM measure_schema -- ISO 10303-41
(derived_unit);

REFERENCE FROM person_organization_schema -- ISO 10303-41
(address,
 person_and_organization_role,
 person_and_organization,
 person_role,
 organization_role,
 organizational_project);

REFERENCE FROM product_definition_schema -- ISO 10303-41
(product_category,
 product_definition,
 product_definition_substitute);

REFERENCE FROM product_property_definition_schema -- ISO 10303-41
(property_definition,
 shape_aspect,
 shape_aspect_relationship);

REFERENCE FROM product_property_representation_schema -- ISO 10303-41
(context_dependent_shape_representation,
 property_definition_representation);

REFERENCE FROM group_schema -- ISO 10303-41
(group);

REFERENCE FROM representation_schema -- ISO 10303-43
(representation);

REFERENCE FROM support_resource_schema -- ISO 10303-41
(identifier,
 label,
 text);
(*)

```

NOTE 1 The schema reference above are specified in the following parts of ISO 10303:

action_schema	clause 10 of this part of ISO 10303
application_context_schema	clause 4 of this part of ISO 10303
approval_schema	clause 12 of this part of ISO 10303
configuration_management_schema	ISO 10303-44
date_time_schema	clause 16 of this part of ISO 10303
effectivity_schema	clause 18 of this part of ISO 10303
external_reference_schema	clause 19 of this part of ISO 10303
management_resources_schema	clause 8 of this part of ISO 10303
measure_schema	clause 21 of this part of ISO 10303
person_organization_schema	clause 15 of this part of ISO 10303
product_definition_schema	clause 5 of this part of ISO 10303
product_property_definition_schema	clause 6 of this part of ISO 10303
product_property_representation_schema	clause 7 of this part of ISO 10303
representation_schema	ISO 10303-43
support_resource_schema	clause 20 of this part of ISO 10303

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

NOTE 4 This schema contains support resources.

22.1 Introduction

The subject of the **basic_attribute_schema** is the assignment of id, name, description and role attributes to aspects of product data. This schema defines the mechanism to define a value for these attributes and associate it with product data.

NOTE 1 The concepts defined in this schema do not provide a generic capability to associate an identification, a name, a description, or a role to product data. They are applied, in order to ensure upward compatibility, in the case where a requirement to add one or more of these kinds of attributes to a particular entity data type has been identified during the revision or amendment process of the standard document in which this entity data type is defined.

NOTE 2 The associated entities are defined in ISO 10303-41, ISO 10303-43 and ISO 10303-44.

22.2 Fundamental concepts and assumptions

The following concepts are supported by this schema:

- the assignment of an identifier to product data;
- the assignment of a label to product data;
- the assignment of a descriptive text to product data;
- the assignment of a role to an association of management type data with other aspects of product data.

The assignment of an identifier corresponds to the **id_attribute** entity data type. The entities to which an identifier may be assigned are specified in the **id_attribute_select** data type. These entities have a derived attribute **id**. The corresponding identifier value is provided by the **get_id_value** function. A constraint ensures that there is at most one instance of **id_attribute** assigned to an instance of an entity in the **id_attribute_select** type.

The assignment of a name corresponds to the **name_attribute** entity data type. The entities to which a name may be assigned are specified in the **name_attribute_select** data type. These entities have a derived attribute **name**. The corresponding label value is provided by the **get_name_value** function. A constraint ensures that there is at most one instance of **name_attribute** assigned to an instance of an entity in the **name_attribute_select** type.

The assignment of a descriptive text corresponds to the **description_attribute** entity data type. The entities to which a description may be assigned are specified in the **description_attribute_select** data type. These entities have a derived attribute **description**. The corresponding text value is provided by the **get_description_value** function. A constraint ensures that there is at most one instance of **description_attribute** assigned to an instance of an entity in the **description_attribute_select** type.

The assignment of a role corresponds to the **role_association** entity data type. The entities to which a role may be assigned are specified in the **role_select** data type. These entities have a derived attribute **role**. The corresponding **object_role** is provided by the **get_role** function. A constraint ensures that there is at most one instance of **role_association** assigned to an instance of an entity in the **role_select** type.

22.3 Basic attribute type definitions

This subclause contains the EXPRESS type definitions in the **basic_attribute_schema**.

22.3.1 description_attribute_select

A **description_attribute_select** specifies those objects that may be assigned a descriptive text.

EXPRESS specification

```
* )
TYPE description_attribute_select = SELECT
  (action_request_solution,
   application_context,
   approval_role,
   configuration_design,
   date_role,
   date_time_role,
   context_dependent_shape_representation,
```

```
    effectivity,  
    external_source,  
    organization_role,  
    person_and_organization_role,  
    person_and_organization,  
    person_role,  
    property_definition_representation,  
    representation,  
    time_role);  
END_TYPE; -- description_attribute_select  
(*
```

22.3.2 id_attribute_select

An **id_attribute_select** specifies those objects that may be assigned an identifier.

EXPRESS specification

```
*)  
TYPE id_attribute_select = SELECT  
    (action,  
     address,  
     product_category,  
     property_definition,  
     shape_aspect,  
     shape_aspect_relationship,  
     application_context,  
     group,  
     organizational_project,  
     representation);  
END_TYPE; -- id_attribute_select  
(*
```

22.3.3 name_attribute_select

A **name_attribute_select** specifies those objects that may be assigned a name.

EXPRESS specification

```
*)  
TYPE name_attribute_select = SELECT  
    (action_request_solution,  
     address,  
     configuration_design,  
     context_dependent_shape_representation,  
     derived_unit,  
     effectivity,  
     person_and_organization,  
     product_definition,  
     product_definition_substitute,  
     property_definition_representation);  
END_TYPE; -- name_attribute_select  
(*
```

22.3.4 role_select

A **role_select** specifies those objects that may be assigned a role.

EXPRESS specification

```

*)
TYPE role_select = SELECT
  (action_assignment,
   action_request_assignment,
   approval_assignment,
   approval_date_time,
   certification_assignment,
   contract_assignment,
   document_reference,
   effectivity_assignment,
   external_referent_assignment,
   group_assignment,
   name_assignment,
   security_classification_assignment);
END_TYPE; -- role_select
(*)

```

22.4 Basic attribute entity definitions

This subclause contains the EXPRESS entity definitions in the **basic_attribute_schema**.

22.4.1 description_attribute

A **description_attribute** specifies the assignment of a descriptive text string to product data.

NOTE This entity provides the capability to add a description attribute to product data using a method that is upwardly compatible with ISO 10303-41:1994, ISO 10303-43:1994, and ISO 10303-44:1994.

EXPRESS specification

```

*)
ENTITY description_attribute;
  attribute_value : text;
  described_item : description_attribute_select;
END_ENTITY; -- description_attribute
(*)

```

Attribute definitions

attribute_value: the text that characterizes the **described_item**.

described_item: the item that is described.

22.4.2 id_attribute

An **id_attribute** is the assignment of an identifier to product data.

NOTE 1 This entity provides the capability to add an id attribute to product data using a method that is upwardly compatible with ISO 10303-41:1994 and ISO 10303-43:1994.

EXPRESS specification

```

*)
ENTITY id_attribute;
  attribute_value : identifier;
  identified_item : id_attribute_select;

```

ISO 10303-41:2000(E)

```
END_ENTITY; -- id_attribute
( *
```

Attribute definitions

attribute_value: the **identifier** that distinguishes the **identified_item**.

NOTE 2 The context in which **attribute_value** is used as a discriminating characteristic can be identified in an annotated Express schema that uses or specializes this entity, or by default, in an agreement of common understanding between partners sharing this information.

identified_item: the item that is identified.

22.4.3 name_attribute

A **name_attribute** is the assignment of a label by which the product data is known.

NOTE This entity provides the capability to add a name attribute to product data using a method that is upwardly compatible with ISO 10303-41:1994 and ISO 10303-44:1994.

EXPRESS specification

```
*)
ENTITY name_attribute;
  attribute_value : label;
  named_item      : name_attribute_select;
END_ENTITY; -- name_attribute
( *
```

Attribute definitions

attribute_value: the label by which the **named_item** is known.

named_item: the item to which the name is applied.

22.4.4 object_role

An **object_role** specifies a role for the association of management type data with other aspects of product data and a description of that role.

NOTE This entity provides the capability to add a role attribute to an association of management type data with other aspects of product data using a method that is upwardly compatible with ISO 10303-41:1994.

EXPRESS specification

```
*)
ENTITY object_role;
  name          : label;
  description    : OPTIONAL text;
END_ENTITY; -- object_role
( *
```

Attribute definitions

name: the **label** by which the **object_role** is known.

description: the **text** that characterizes the **object_role**. The value of the attribute need not be specified.

22.4.5 role_association

A **role_association** is the assignment of an **object_role** to an association of management type data with other aspects of product data.

NOTE This entity data type provides a capability to add a role attribute to certain entity data types that specify associations between product data and administrative or management data. Such entity data types are specified in the **management_resources_schema**.

EXPRESS specification

```
*)
ENTITY role_association;
  role      : object_role;
  item_with_role : role_select;
END_ENTITY; -- role_association
(*
```

Attribute definitions

role: the **object_role** that specifies the purpose of the association of the **role_association** with product_data.

item_with_role: the item to which a role is assigned.

22.5 Basic attribute function definitions

This subclause contains the EXPRESS function definitions in the **basic_attribute_schema**.

22.5.1 get_description_value

The function **get_description_value** accepts an object to which a descriptive text string may be assigned and returns this description if present.

If a **description_attribute** references the object, the function will return the value of the **attribute_value** attribute of this **description_attribute**. Otherwise, the indeterminate value will be returned.

EXPRESS specification

```
*)
FUNCTION get_description_value
  (obj : description_attribute_select) : text;
LOCAL
  description_bag : BAG OF description_attribute :=
    (USEDIN (obj,
             'BASIC_ATTRIBUTE_SCHEMA.' +
             'DESCRIPTION_ATTRIBUTE.' +
             'DESCRIBED_ITEM'));
END_LOCAL;
IF SIZEOF (description_bag) = 1
  THEN RETURN (description_bag[1].attribute_value);
  ELSE RETURN (?);
END_IF;
END_FUNCTION; -- get_description_value
(*
```

Argument definitions:

obj: (input) the object for which a **text** is determined. This is an input to the function.

22.5.2 get_id_value

The function **get_id_value** accepts an object to which an identifier may be assigned and returns this identifier if present.

If an **id_attribute** references the object, the function will return the value of the **attribute_value** attribute of this **id_attribute**. Otherwise, the indeterminate value will be returned.

EXPRESS specification

```

*)
FUNCTION get_id_value
  (obj : id_attribute_select) : identifier;
  LOCAL
    id_bag : BAG OF id_attribute :=
      (USEDIN (obj,
        'BASIC_ATTRIBUTE_SCHEMA.' +
        'ID_ATTRIBUTE.' +
        'IDENTIFIED_ITEM'));
  END_LOCAL;
  IF SIZEOF (id_bag) = 1
    THEN RETURN (id_bag[1].attribute_value);
    ELSE RETURN (?);
  END_IF;
END_FUNCTION; -- get_id_value
(*

```

Argument definitions:

obj: (input) the object for which an **identifier** is determined. This is an input to the function.

22.5.3 get_name_value

The function **get_name_value** accepts an object to which a label may be assigned and returns this label if present.

If a **name_attribute** references the object, the function will return the value of the **attribute_value** attribute of this **name_attribute**. Otherwise, the indeterminate value will be returned.

EXPRESS specification

```

*)
FUNCTION get_name_value
  (obj : name_attribute_select) : label;
  LOCAL
    name_bag : BAG OF name_attribute :=
      (USEDIN (obj,
        'BASIC_ATTRIBUTE_SCHEMA.' +
        'NAME_ATTRIBUTE.' +
        'NAMED_ITEM'));
  END_LOCAL;
  IF SIZEOF (name_bag) = 1
    THEN RETURN (name_bag[1].attribute_value);
    ELSE RETURN (?);
  END_IF;

```

```
END_FUNCTION; -- get_name_value
(*
```

Argument definitions:

obj: the object for which a **label** is determined. This is an input to the function.

22.5.4 get_role

The function **get_role** accepts an object to which a role may be assigned and returns this role if present. If a **role_association** references the object, the function will return the **object_role** of the role attribute of this **role_association**. Otherwise, the indeterminate value will be returned.

EXPRESS specification

```
*)
FUNCTION get_role
  (obj : role_select) : object_role;
  LOCAL
    role_bag : BAG OF role_association:=
      (USEDIN (obj,
        'BASIC_ATTRIBUTE_SCHEMA.' +
        'ROLE_ASSOCIATION.' +
        'ITEM_WITH_ROLE'));
  END_LOCAL;
  IF SIZEOF (role_bag) = 1
    THEN RETURN (role_bag[1].role);
    ELSE RETURN (?);
  END_IF;
END_FUNCTION; -- get_role
(*
```

Argument definitions:

obj: (input) the object for which an **object_role** is determined. This is an input to the function.

EXPRESS specification

```
*)
END_SCHEMA; -- basic_attribute_schema
(*
```

Annex A (normative)

Short names of entities

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

NOTE The EXPRESS entity names are available from Internet:

<http://www.mel.nist.gov/div826/subject/apde/snr/>

Table A.1 - Short names of entities

Entity names	Short names
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_DIRECTIVE	ACTDRC
ACTION_METHOD	ACTMTH
ACTION_METHOD_ASSIGNMENT	ACMTAS
ACTION_METHOD_RELATIONSHIP	ACMTRL
ACTION_METHOD_ROLE	ACM0
ACTION_RELATIONSHIP	ACTRLT
ACTION_REQUEST_ASSIGNMENT	ACRQAS
ACTION_REQUEST_SOLUTION	ACRQSL
ACTION_REQUEST_STATUS	ACRQST
ACTION_RESOURCE	ACTRSR
ACTION_RESOURCE_RELATIONSHIP	ACRSRL
ACTION_RESOURCE_TYPE	ACRSTY
ACTION_STATUS	ACTSTT
ADDRESS	ADDRSS
AMOUNT_OF_SUBSTANCE_MEASURE_WITH_UNIT	AOSMWU
AMOUNT_OF_SUBSTANCE_UNIT	AOSU
APPLICATION_CONTEXT	APPCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
APPLICATION_CONTEXT_RELATIONSHIP	APCNRL
APPLICATION_PROTOCOL_DEFINITION	APPRDF
APPROVAL	APPRVL
APPROVAL_ASSIGNMENT	APPASS
APPROVAL_DATE_TIME	APDTTM
APPROVAL_PERSON_ORGANIZATION	APPROR
APPROVAL_RELATIONSHIP	APPRLT

Table A.1 - Short names of entities (continued)

Entity names	Short names
APPROVAL_ROLE	APPRL
APPROVAL_STATUS	APPSTT
AREA_MEASURE_WITH_UNIT	AMWU
AREA_UNIT	ARUNT
ATTRIBUTE_CLASSIFICATION_ASSIGNMENT	ATCLAS
ATTRIBUTE_VALUE_ASSIGNMENT	ATVLAS
ATTRIBUTE_VALUE_ROLE	ATVLRL
CALENDAR_DATE	CLNDT
CELSIUS_TEMPERATURE_MEASURE_WITH_UNIT	CTMWU
CERTIFICATION	CRTECT
CERTIFICATION_ASSIGNMENT	CRTASS
CERTIFICATION_TYPE	CRTTYP
CHARACTERIZED_OBJECT	CHROBJ
CHARACTERIZED_OBJECT_RELATIONSHIP	CHOBRL
CLASSIFICATION_ASSIGNMENT	CLSASS
CLASSIFICATION_ROLE	CLSRL
CONTEXT_DEPENDENT_SHAPE_REPRESENTATION	CDSR
CONTEXT_DEPENDENT_UNIT	CNDPUN
CONTRACT	CNTRCT
CONTRACT_ASSIGNMENT	CNTASS
CONTRACT_RELATIONSHIP	CNTRLT
CONTRACT_TYPE	CNTTYP
CONVERSION_BASED_UNIT	CNBSUN
COORDINATED_UNIVERSAL_TIME_OFFSET	CUTO
DATE	DATE
DATED_EFFECTIVITY	DTDEFF
DATE_AND_TIME	DTANTM
DATE_AND_TIME_ASSIGNMENT	DATA
DATE_ASSIGNMENT	DTASS
DATE_ROLE	DTRL
DATE_TIME_ROLE	DTMRL
DERIVED_UNIT	DRVUNT
DERIVED_UNIT_ELEMENT	DRUNEL
DESCRIPTION_ATTRIBUTE	DSCATT
DIMENSIONAL_EXPONENTS	DMNEXP
DIRECTED_ACTION	DRCACT
DOCUMENT	DCMNT
DOCUMENT_PRODUCT_ASSOCIATION	DCPI
DOCUMENT_REFERENCE	DCMRFR
DOCUMENT_RELATIONSHIP	DCMRLT
DOCUMENT_REPRESENTATION_TYPE	DCRPTY

Table A.1 - Short names of entities (continued)

Entity names	Short names
DOCUMENT_TYPE	DCMTYP
DOCUMENT_USAGE_CONSTRAINT	DCUSCN
DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	DUCA
DOCUMENT_USAGE_ROLE	DCUSRL
DOCUMENT_WITH_CLASS	DCWTCL
EFFECTIVITY	EFFCTV
EFFECTIVITY_ASSIGNMENT	EFFASS
EFFECTIVITY_CONTEXT_ASSIGNMENT	EFCO
EFFECTIVITY_CONTEXT_ROLE	EFCNRL
EFFECTIVITY_RELATIONSHIP	EFFRLT
ELECTRIC_CURRENT_MEASURE_WITH_UNIT	ECMWU
ELECTRIC_CURRENT_UNIT	ELCRUN
EVENT_OCCURRENCE	EVNOCC
EVENT_OCCURRENCE_ASSIGNMENT	EVOCAS
EVENT_OCCURRENCE_CONTEXT_ASSIGNMENT	EOCA
EVENT_OCCURRENCE_CONTEXT_ROLE	EOCR
EVENT_OCCURRENCE_RELATIONSHIP	EVOO
EVENT_OCCURRENCE_ROLE	EVOCRL
EXECUTED_ACTION	EXCACT
EXTERNALLY_DEFINED_ITEM	EXDFIT
EXTERNALLY_DEFINED_ITEM_RELATIONSHIP	EDIR
EXTERNAL_IDENTIFICATION_ASSIGNMENT	EXIDAS
EXTERNAL_REFERENT_ASSIGNMENT	EXRFAS
EXTERNAL_SOURCE	EXTSRC
EXTERNAL_SOURCE_RELATIONSHIP	EXSRRL
GENERAL_PROPERTY	GNRPRP
GENERAL_PROPERTY_ASSOCIATION	GNPRAS
GENERAL_PROPERTY_RELATIONSHIP	GNPRRL
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
GROUP	GROUP
GROUP_ASSIGNMENT	GRPASS
GROUP_RELATIONSHIP	GRPRLT
IDENTIFICATION_ASSIGNMENT	IDNASS
IDENTIFICATION_ASSIGNMENT_RELATIONSHIP	IDASRL
IDENTIFICATION_ROLE	IDNRL
ID_ATTRIBUTE	IDATT
ITEM_IDENTIFIED_REPRESENTATION_USAGE	IIRU
LENGTH_MEASURE_WITH_UNIT	LMWU
LENGTH_UNIT	LNGUNT
LIBRARY_ASSIGNMENT	LBRASS
LIBRARY_CONTEXT	LBRCNT

Table A.1 - Short names of entities (continued)

Entity names	Short names
LOCAL_TIME	LCLTM
LOT_EFFECTIVITY	LTEFF
LUMINOUS_INTENSITY_MEASURE_WITH_UNIT	LIMWU
LUMINOUS_INTENSITY_UNIT	LMINUN
MASS_MEASURE_WITH_UNIT	MMWU
MASS_UNIT	MSSUNT
MEASURE_WITH_UNIT	MSWTUN
NAMED_UNIT	NMDUNT
NAME_ASSIGNMENT	NMASS
NAME_ATTRIBUTE	NMATT
OBJECT_ROLE	OBJRL
ORDINAL_DATE	ORDDT
ORGANIZATION	ORGNZT
ORGANIZATIONAL_ADDRESS	ORGADD
ORGANIZATIONAL_PROJECT	ORGPRJ
ORGANIZATIONAL_PROJECT_ASSIGNMENT	ORPRAS
ORGANIZATIONAL_PROJECT_RELATIONSHIP	ORPO
ORGANIZATIONAL_PROJECT_ROLE	ORPRRL
ORGANIZATION_ASSIGNMENT	ORGASS
ORGANIZATION_RELATIONSHIP	ORGRLT
ORGANIZATION_ROLE	ORGRL
PERSON	PERSON
PERSONAL_ADDRESS	PRSADD
PERSON_AND_ORGANIZATION	PRANOR
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION_ROLE	PAOR
PERSON_ASSIGNMENT	PRSASS
PERSON_ROLE	PRSRL
PLANE_ANGLE_MEASURE_WITH_UNIT	PAMWU
PLANE_ANGLE_UNIT	PLANUN
PRE_DEFINED_ITEM	PRDFIT
PRODUCT	PRDCT
PRODUCT_CATEGORY	PRDCTG
PRODUCT_CATEGORY_RELATIONSHIP	PRCTRL
PRODUCT_CONCEPT_CONTEXT	PRCNCN
PRODUCT_CONTEXT	PRDCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_CONTEXT	PRDFCN
PRODUCT_DEFINITION_CONTEXT_ASSOCIATION	PDCA
PRODUCT_DEFINITION_CONTEXT_ROLE	PDCR
PRODUCT_DEFINITION_EFFECTIVITY	PRDFEF

Table A.1 - Short names of entities (continued)

Entity names	Short names
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_FORMATION_RELATIONSHIP	PDFR
PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE	PDFWSS
PRODUCT_DEFINITION_RELATIONSHIP	PRDFRL
PRODUCT_DEFINITION_SHAPE	PRDFSH
PRODUCT_DEFINITION_SUBSTITUTE	PRDFSB
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	PDWAD
PRODUCT_RELATED_PRODUCT_CATEGORY	PRPC
PRODUCT_RELATIONSHIP	PRDRLT
PROPERTY_DEFINITION	PRPDFN
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
RATIO_MEASURE_WITH_UNIT	RMWU
RATIO_UNIT	RTUNT
RELATIVE_EVENT_OCCURRENCE	RLEVOC
ROLE_ASSOCIATION	RLASS
SECURITY_CLASSIFICATION	SCRCLS
SECURITY_CLASSIFICATION_ASSIGNMENT	SCCLAS
SECURITY_CLASSIFICATION_LEVEL	SCCLLV
SERIAL_NUMBERED_EFFECTIVITY	SRNMEF
SHAPE_ASPECT	SHPASP
SHAPE_ASPECT_RELATIONSHIP	SHASRL
SHAPE_DEFINITION_REPRESENTATION	SHDFRP
SHAPE_REPRESENTATION	SHPRPR
SHAPE_REPRESENTATION_RELATIONSHIP	SHRPRL
SI_UNIT	SUNT
SOLID_ANGLE_MEASURE_WITH_UNIT	SAMWU
SOLID_ANGLE_UNIT	SLANUN
THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT	TTMWU
THERMODYNAMIC_TEMPERATURE_UNIT	THTMUN
TIME_ASSIGNMENT	TMASS
TIME_INTERVAL	TMINT
TIME_INTERVAL_ASSIGNMENT	TMINAS
TIME_INTERVAL_BASED_EFFECTIVITY	TIBE
TIME_INTERVAL_RELATIONSHIP	TMINRL
TIME_INTERVAL_ROLE	TMIO
TIME_INTERVAL_WITH_BOUNDS	TIWB
TIME_MEASURE_WITH_UNIT	TMWU
TIME_ROLE	TMRL
TIME_UNIT	TMUNT

Table A.1 - Short names of entities (concluded)

Entity names	Short names
VERSIONED_ACTION_REQUEST	VRACRQ
VOLUME_MEASURE_WITH_UNIT	VMWU
VOLUME_UNIT	VLMUNT
WEEK_OF_YEAR_AND_DAY_DATE	WOYADD

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-41:2000

Annex B
(normative)

Information object registration

B.1 Document identification

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

{ iso standard 10303 part(41) version(3) }

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

B.2 Schema identification

B.2.1 application_context_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) application-context-schema(1) }

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

B.2.2 product_definition_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) product-definition-schema(2) }

is assigned to the **product_definition_schema** schema (see clause 5). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.3 product_property_definition_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) product-property-definition-schema(3) }

is assigned to the **product_property_definition_schema** schema (see clause 6). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.4 product_property_representation_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) product-property-representation-schema(4) }

is assigned to the **product_property_representation_schema** schema (see clause 7). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.5 management_resources_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) management-resource-schema(5) }

is assigned to the **management_resource_schema** schema (see clause 8). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.6 document_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) document-schema(6) }

is assigned to the **document_schema** schema (see clause 9). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.7 action_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) action-schema(7) }

is assigned to the **action_schema** schema (see clause 10). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.8 certification_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) certification-schema(8) }

is assigned to the **certification-schema** schema (see clause 11). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.9 approval_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) approval-schema(9) }

is assigned to the **approval_schema** schema (see clause 12). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.10 contract_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) contract-schema(10) }

is assigned to the **contract_schema** schema (see clause 13). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.11 security_classification_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) security-classification-schema(11) }

is assigned to the **security_classification_schema** schema (see clause 14). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.12 person_organization_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) person-organization-schema(12) }

is assigned to the **person_organization_schema** schema (see clause 15). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.13 date_time_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) date-time-schema(13) }

is assigned to the **date_time_schema** schema (see clause 16). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.14 group_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) group-schema(14) }

is assigned to the **group_schema** schema (see clause 17). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.15 effectivity_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) effectivity-schema(15) }

is assigned to the **effectivity_schema** schema (see clause 18). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.16 external_reference_schema identification

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

{ iso standard 10303 part(41) version(3) object(1) external-reference-schema(16) }

is assigned to the **external_reference_schema** schema (see clause 19). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.17 support_resource_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) support-resource-schema(17) }

is assigned to the **support_resource_schema** schema (see clause 20). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.18 measure_schema identification

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

{ iso standard 10303 part(41) version(2) object(1) measure-schema(18) }

is assigned to the **measure_schema** schema (see clause 21). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.19 basic_attribute_schema identification

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

{ iso standard 10303 part(41) version(1) object(1) basic-attribute-schema(19) }

is assigned to the **basic_attribute_schema** schema (see clause 22). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-41:2000

Annex C (informative)

Computer-interpretable listing

This annex references a listing of the EXPRESS entity names and corresponding short names as specified in this part of ISO 10303. It also references a listing of each EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text. These listings are available in computer-interpretable form and can be found at the following URLs:

Short names: <<http://www.mel.nist.gov/div826/subject/apde/snr/>>
EXPRESS: <<http://www.mel.nist.gov/step/parts/part041/ed2/fdis/>>

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

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

Annex D
(informative)

EXPRESS-G diagrams

Figures D.1 through D.44 correspond to the EXPRESS schemas specified in this part of ISO 10303. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-41:2000

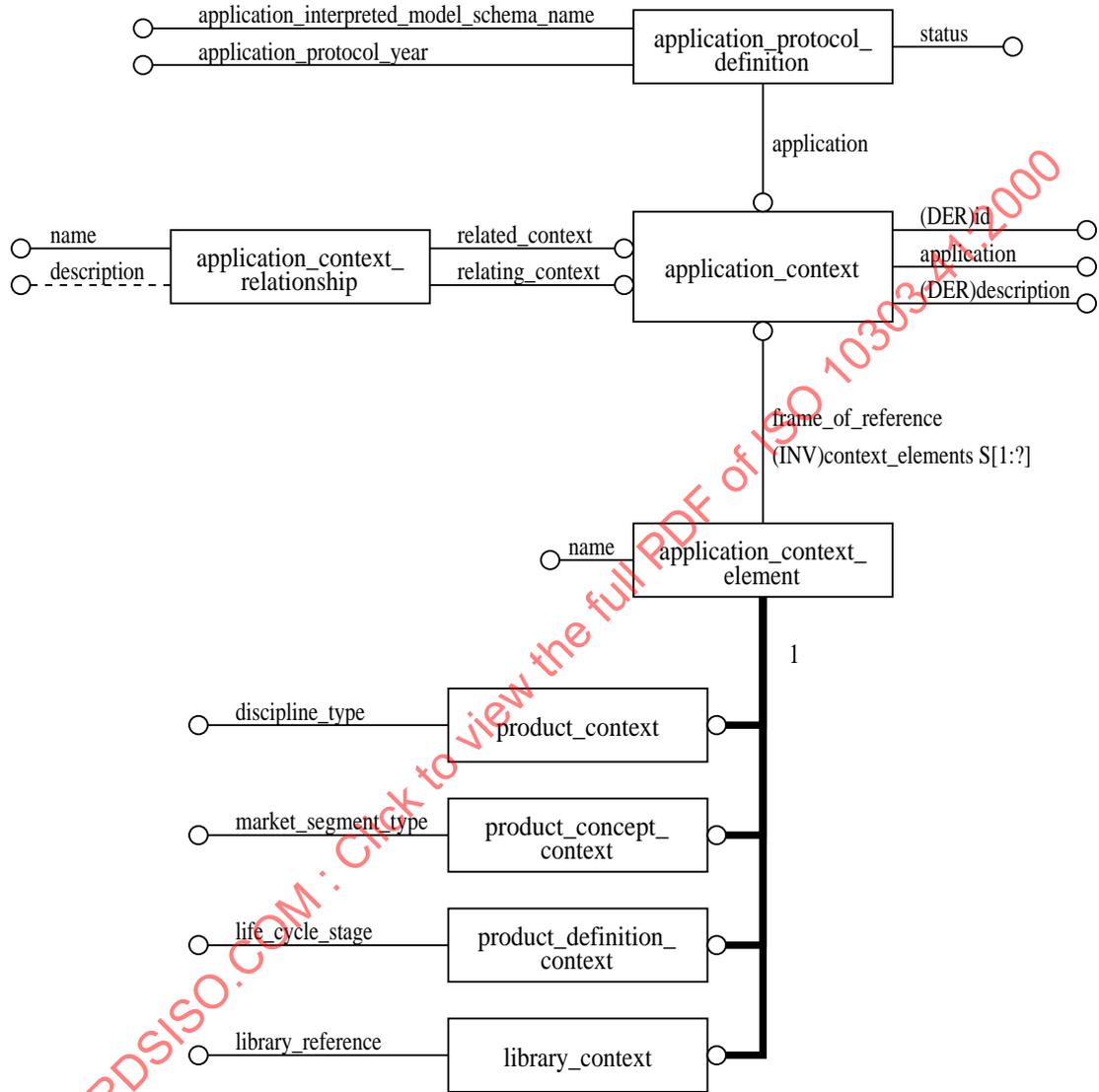


Figure D.1 - EXPRESS-G diagram of the application_context_schema (1 of 1)

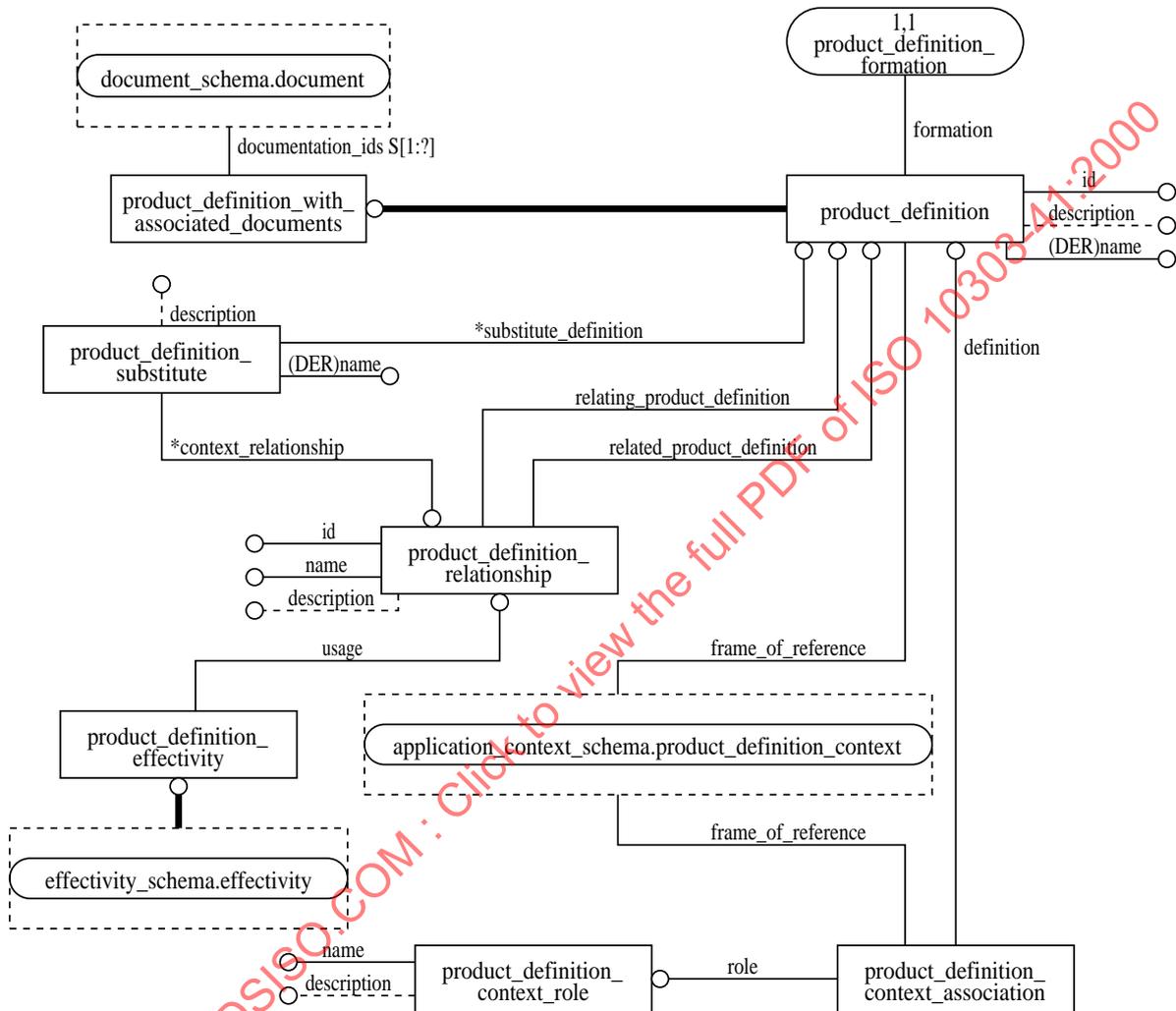


Figure D.3 - EXPRESS-G diagram of the product_definition_schema (2 of 2)

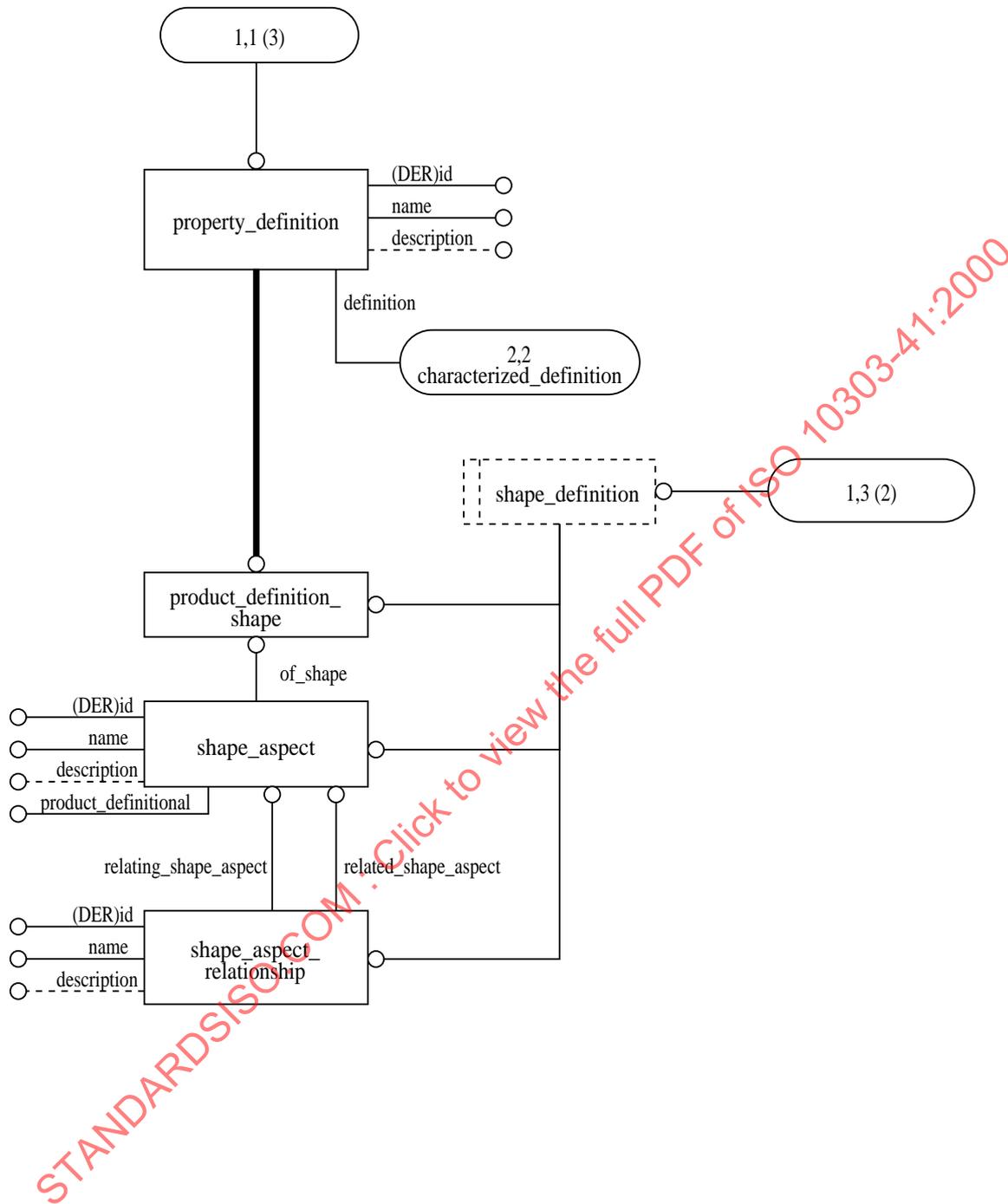


Figure D.4 - EXPRESS-G diagram of the product_property_definition - schema (1 of 3)

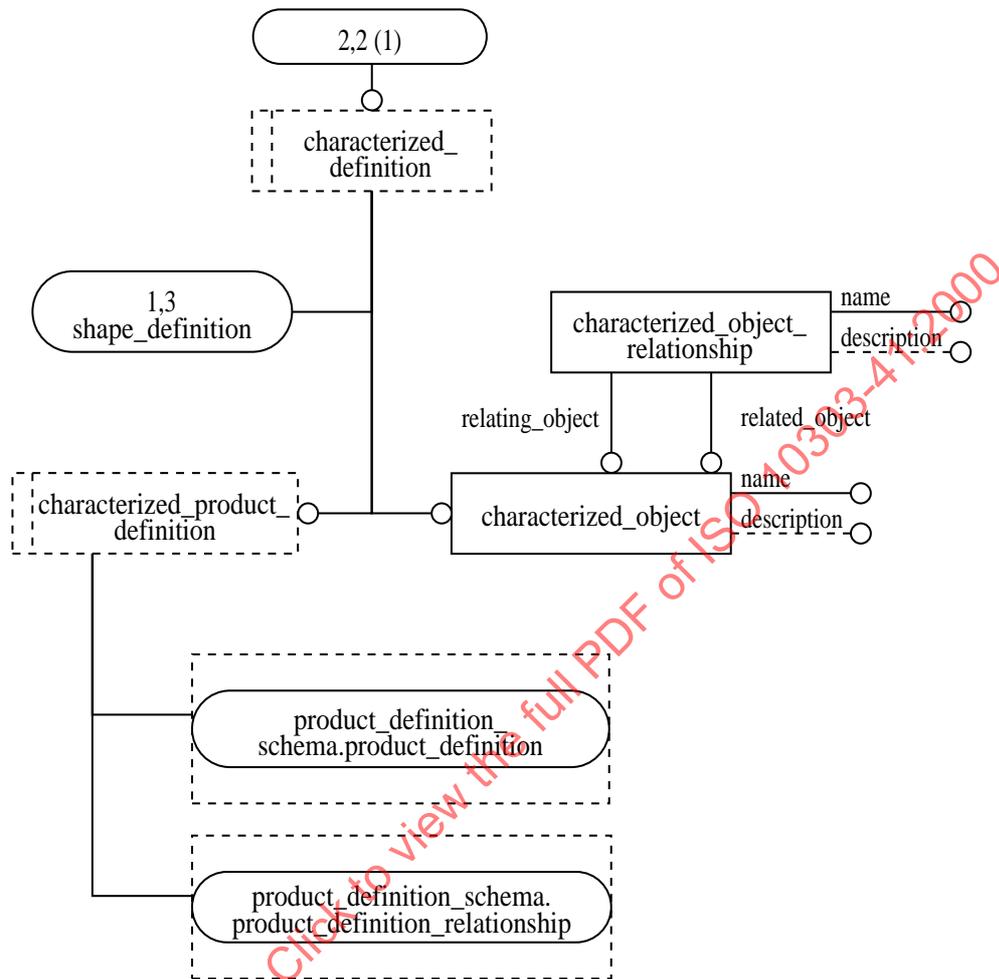
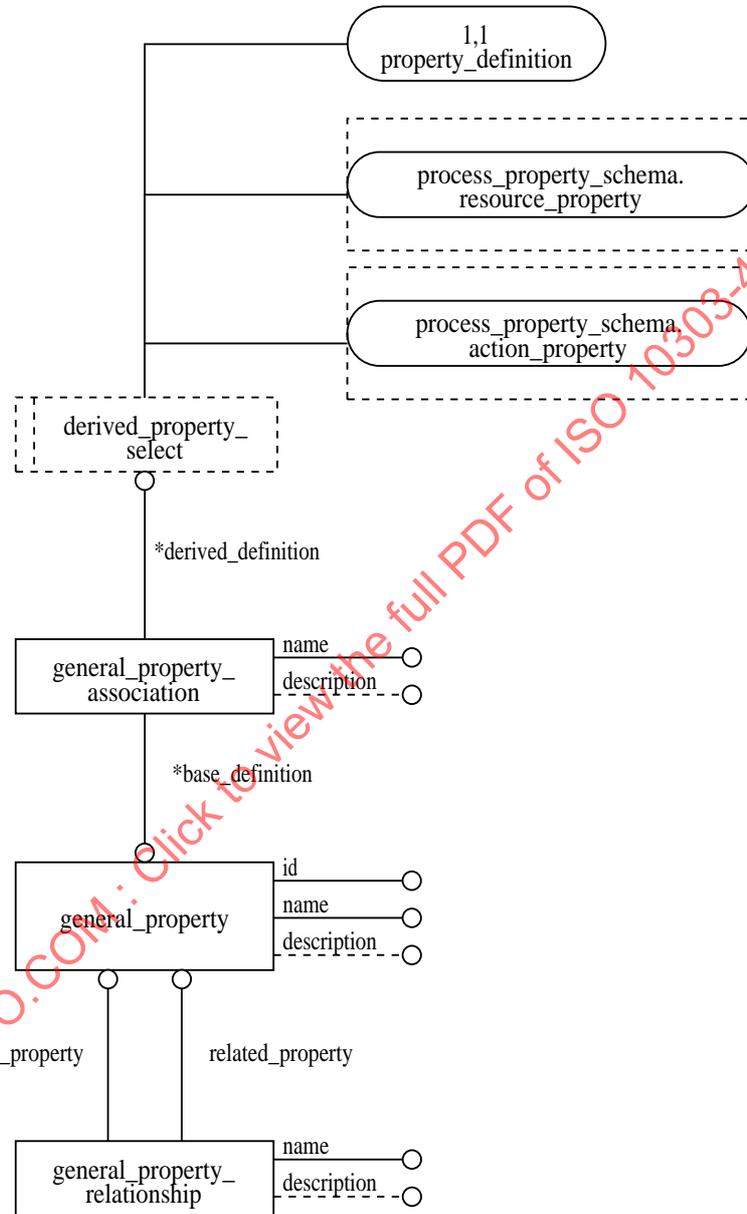


Figure D.5 - EXPRESS-G diagram of the product_property_definition_ - schema (2 of 3)



STANDARDS ISO.COM: Click to view the full PDF of ISO 10303-41:2000

Figure D.6 - EXPRESS-G diagram of the product_property_definition_ - schema (3 of 3)

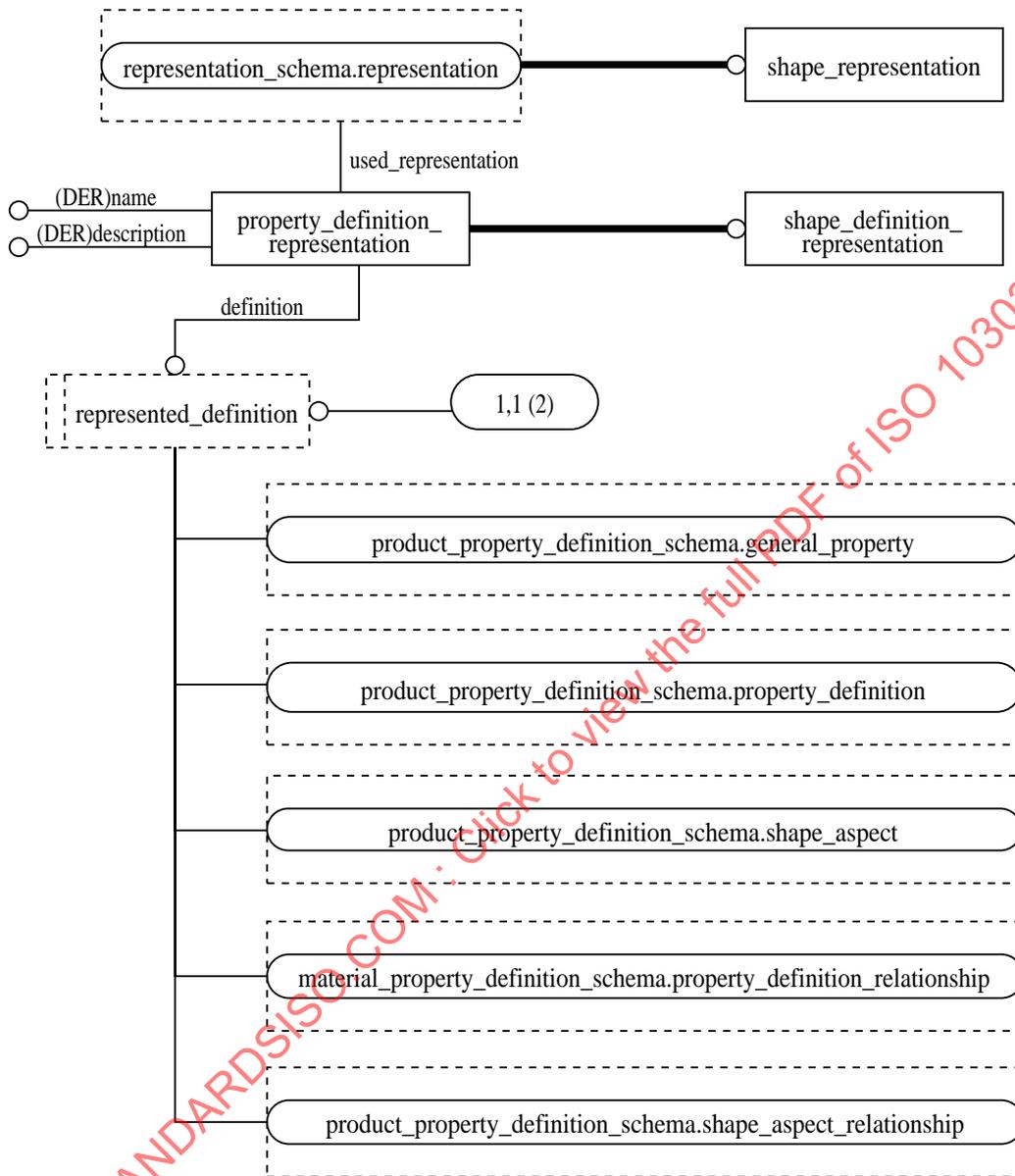


Figure D.7 - EXPRESS-G diagram of the product_property_definition_representation_schema (1 of 2)

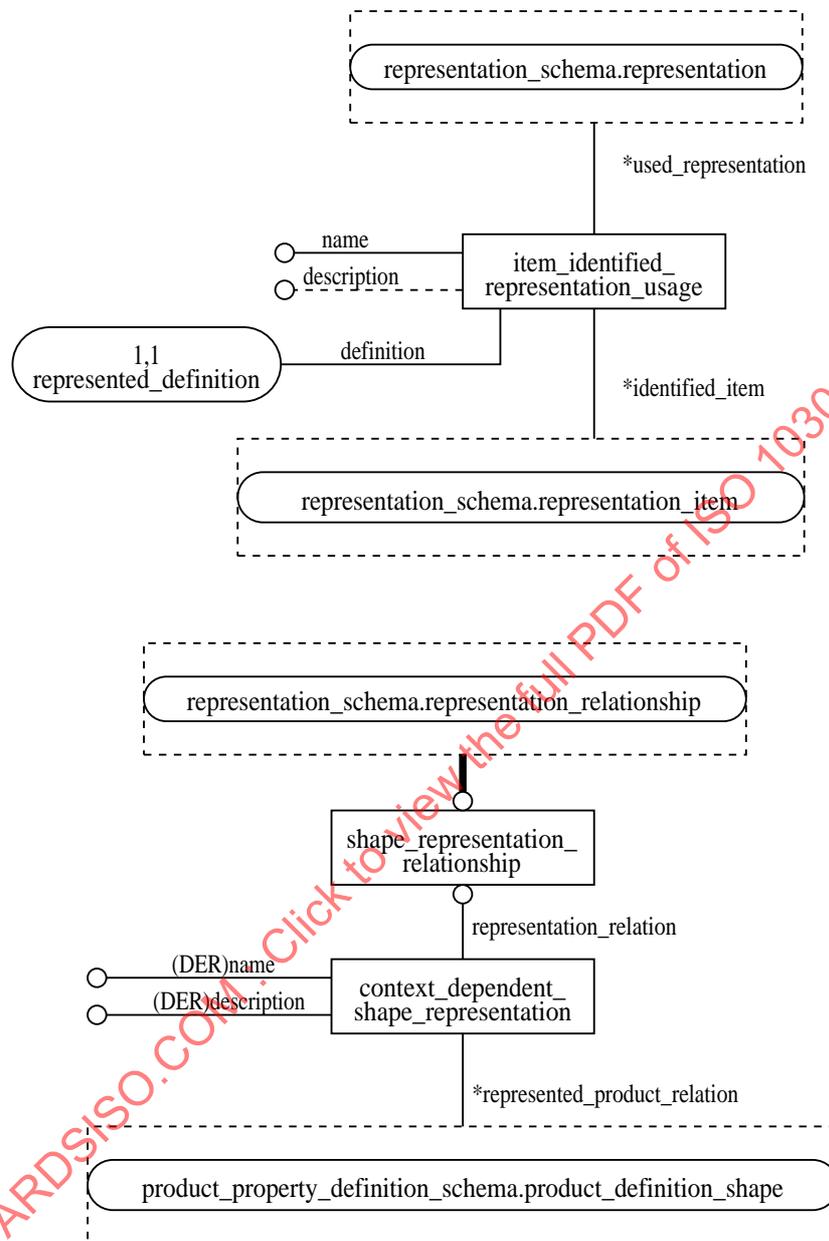


Figure D.8 - EXPRESS-G diagram of the product_property_representation_schema (2 of 2)

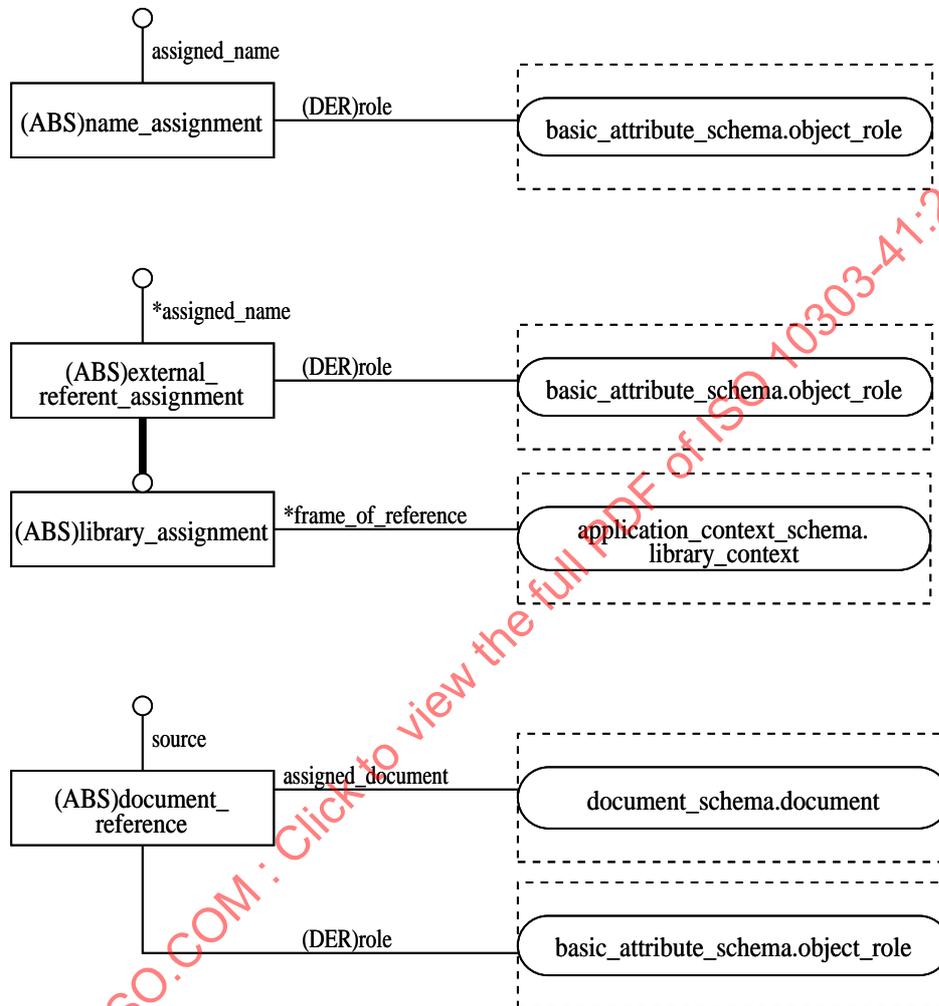


Figure D.9 - EXPRESS-G diagram of the management_resources_schema (1 of 8)

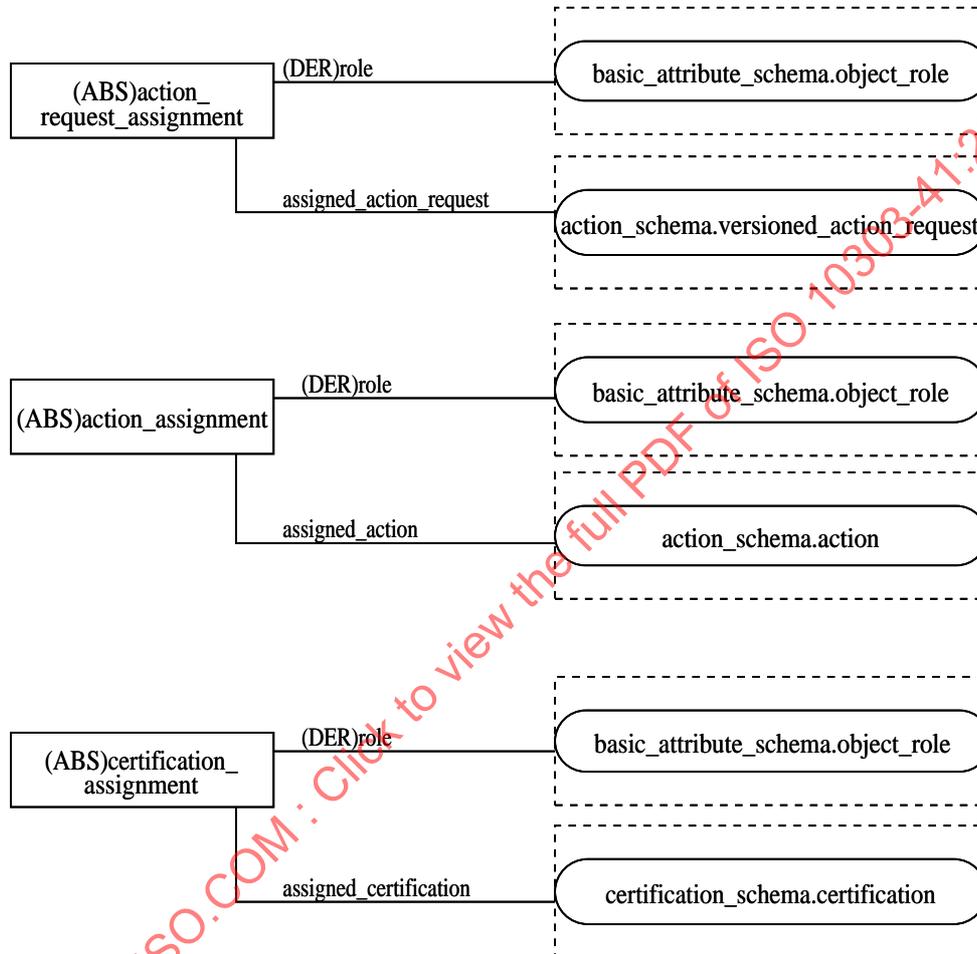


Figure D.10 - EXPRESS-G diagram of the management_resources_schema (2 of 8)

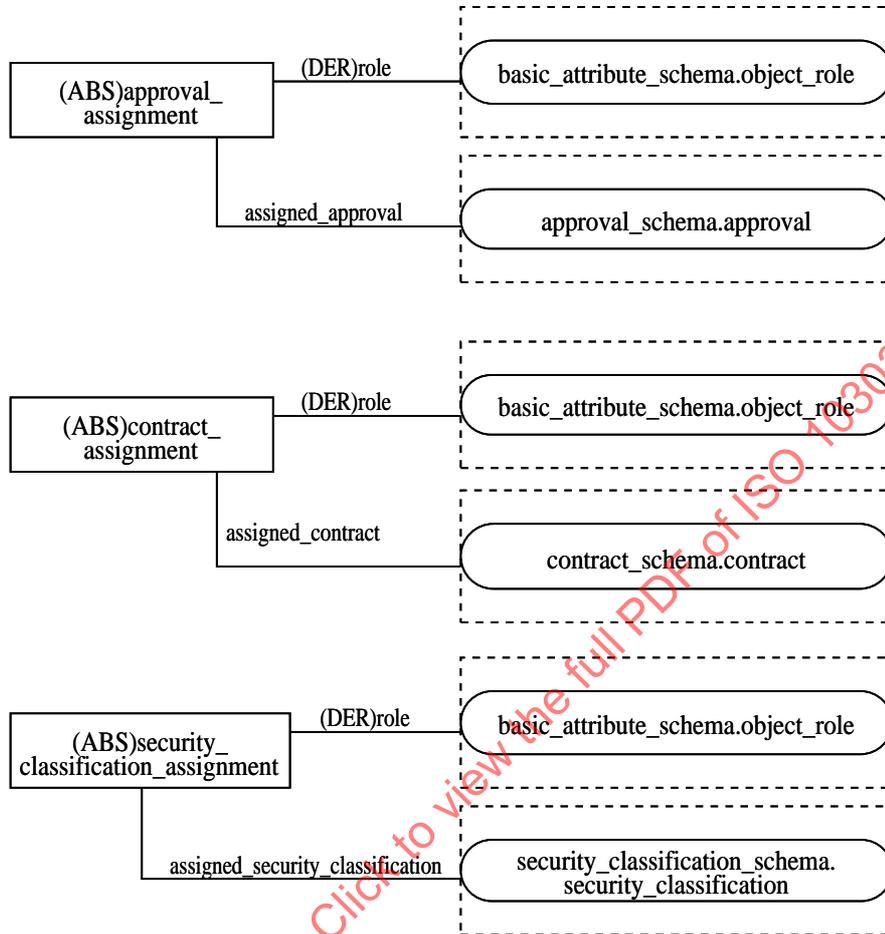


Figure D.11 - EXPRESS-G diagram of the management_resources_schema (3 of 8)

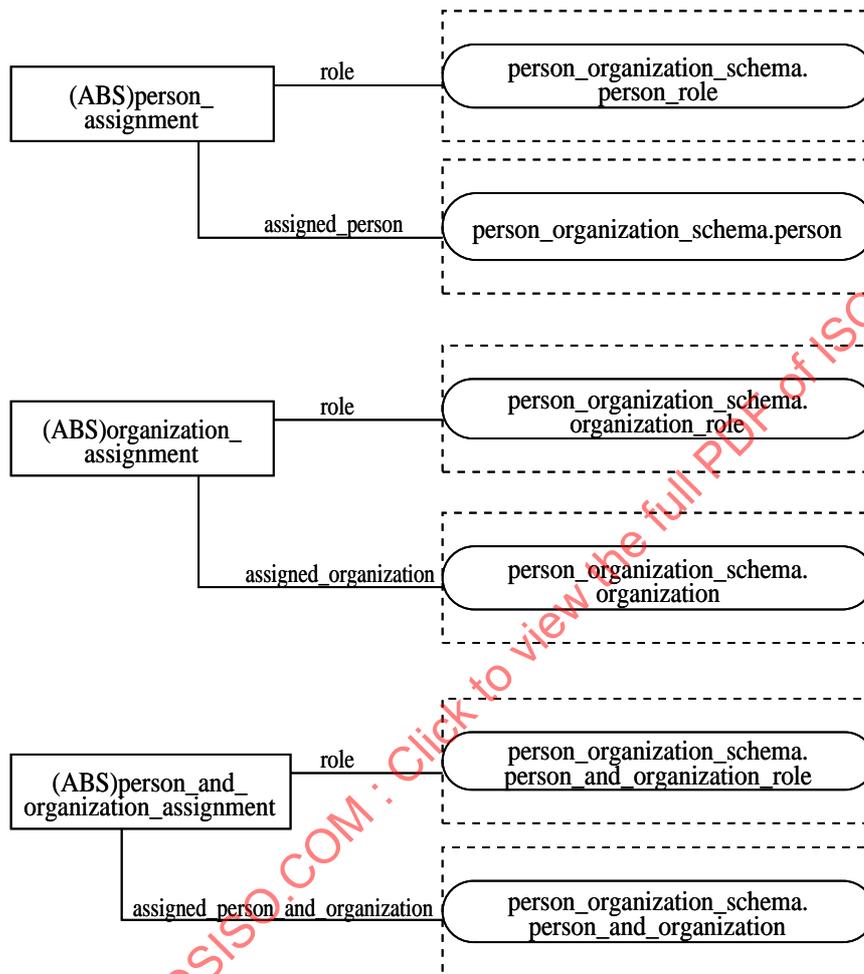


Figure D.12 - EXPRESS-G diagram of the management_resources_schema (4 of 8)