INTERNATIONAL STANDARD

ISO 10303-41

Sixth edition

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

Reference number
ISO 10303-41:2019(E)

© ISO 2019



COPYRIGHT PROTECTED DOCUMENT

© ISO 2019

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

1 Scope ... **1**

2 Normative references ... **1**

3 Terms and definitions ... **1**

4 Twenty-four schemas to follow! ... **1**

5 action_schema ... **1**

5.1 General ... 1

5.2 Fundamental concepts and assumptions ... 2

5.3 action_schema type definitions ... 2

5.3.1 as_description_attribute_select ... 2

5.3.2 as_id_attribute_select ... 2

5.3.3 as_name_attribute_select ... 2

5.3.4 as_role_select ... 2

5.3.5 supported_item ... 2

5.4 action_schema entity definitions ... 3

5.4.1 action ... 3

5.4.2 action_directive ... 4

5.4.3 action_method ... 4

5.4.4 action_method_relationship ... 5

5.4.5 action_relationship ... 5

5.4.6 action_request_solution ... 6

5.4.7 action_request_status ... 7

5.4.8 action_resource ... 7

5.4.9 action_resource_relationship ... 8

5.4.10 action_resource_type ... 8

5.4.11 action_status ... 9

5.4.12 directed_action ... 9

5.4.13 directed_action_assignment ... 9

5.4.14 executed_action ... 10

5.4.15 versioned_action_request ... 10

5.4.16 versioned_action_request_relationship ... 11

5.4.17 action_directive_relationship ... 12

5.5 action_schema function definitions ... 12

5.5.1 acyclic_action_method_relationship ... 12

5.5.2 acyclic_action_relationship ... 14

5.5.3 acyclic_action_resource_relationship ... 15

5.5.4 acyclic_versioned_action_request_relationship ... 16

5.5.5 acyclic_action_directive_relationship ... 18

6 application_context_schema ... **19**

6.1 General ... 19

6.2 Fundamental concepts and assumptions ... 19

6.3 application_context_schema type definitions ... 20

6.3.1 acs_description_attribute_select ... 20

6.3.2 acs_id_attribute_select ... 20

6.3.3 exchange_identification_context_select ... 20

6.4 application_context_schema entity definitions ... 20

6.4.1 application_context ... 20

6.4.2 application_context_element ... 21

6.4.3 application_context_relationship ... 22

6.4.4 application_protocol_definition ... 23

6.4.5 library_context ... 23

6.4.6 product_concept_context ... 23

6.4.7 product_context ... 24

6.4.8 product_definition_context ... 24

6.4.9 dependent_product_definition_exchange_context ... 25

© ISO 2019 – All rights reserved

6.4.10 identification_dependent_product_definition_exchange_context 26
6.4.11 language_dependent_product_definition_exchange_context 26

7 approval_schema 26
7.1 General 26
7.2 Fundamental concepts and assumptions 27
7.3 approval_schema type definitions 27
7.3.1 apps_description_attribute_select 27
7.3.2 apps_role_select 27
7.4 approval_schema entity definitions 27
7.4.1 approval 27
7.4.2 approval_date_time 28
7.4.3 approval_person_organization 28
7.4.4 approval_relationship 29
7.4.5 approval_role 29
7.4.6 approval_status 30
7.5 approval_schema function definitions 30
7.5.1 acyclic_approval_relationship 30

8 basic_attribute_schema 32
8.1 General 32
8.2 Fundamental concepts and assumptions 32
8.3 basic_attribute_schema type definitions 33
8.3.1 description_attribute_select 33
8.3.2 generic_attribute_list_select 33
8.3.3 generic_attribute_select 33
8.3.4 id_attribute_select 33
8.3.5 name_attribute_select 33
8.3.6 role_select 33
8.4 basic_attribute_schema entity definitions 33
8.4.1 aggregate_id_attribute 33
8.4.2 description_attribute 34
8.4.3 generic_attribute 34
8.4.4 generic_attribute_assignment 35
8.4.5 generic_attribute_list_assignment 35
8.4.6 id_attribute 35
8.4.7 name_attribute 36
8.4.8 object_role 36
8.4.9 role_association 37
8.5 basic_attribute_schema function definitions 37
8.5.1 get_description_value 37
8.5.2 get_id_value 38
8.5.3 get_aggregate_id_value 38
8.5.4 get_name_value 39
8.5.5 get_role 40

9 certification_schema 40
9.1 General 40
9.2 Fundamental concepts and assumptions 41
9.3 certification_schema entity definitions 41
9.3.1 certification 41
9.3.2 certification_type 41

10 contract_schema 42
10.1 General 42
10.2 Fundamental concepts and assumptions 42
10.3 contract_schema entity definitions 42
10.3.1 contract 42
10.3.2 contract_relationship 42
10.3.3 contract_type 43
10.4 contract_schema function definitions 44

 © ISO 2019 – All rights reserved

10.4.1 acyclic_contract_relationship ... 44

11 date_time_schema ... 45
11.1 General ... 45
11.2 Fundamental concepts and assumptions ... 45
11.3 date_time_schema type definitions ... 45
11.3.1 dts_description_attribute_select ... 45
11.3.2 ahead_or_behind ... 46
11.3.3 date_time_or_event_occurrence ... 46
11.3.4 date_time_select ... 46
11.3.5 day_in_month_number ... 46
11.3.6 day_in_week_number ... 46
11.3.7 day_in_year_number ... 47
11.3.8 hour_in_day ... 47
11.3.9 minute_in_hour ... 47
11.3.10 month_in_year_number ... 47
11.3.11 second_in_minute ... 47
11.3.12 week_in_year_number ... 48
11.3.13 year_number ... 48
11.4 date_time_schema entity definitions ... 48
11.4.1 calendar_date ... 48
11.4.2 coordinated_universal_time_offset ... 49
11.4.3 date ... 49
11.4.4 date_and_time ... 50
11.4.5 date_role ... 50
11.4.6 date_time_role ... 51
11.4.7 event_occurrence ... 51
11.4.8 event_occurrence_context_role ... 52
11.4.9 event_occurrence_relationship ... 52
11.4.10 event_occurrence_role ... 53
11.4.11 local_time ... 54
11.4.12 ordinal_date ... 54
11.4.13 relative_event_occurrence ... 55
11.4.14 time_interval ... 55
11.4.15 time_interval_relationship ... 55
11.4.16 time_interval_role ... 56
11.4.17 time_interval_with_bounds ... 57
11.4.18 time_role ... 57
11.4.19 week_of_year_and_day_date ... 58
11.4.20 year_month ... 59
11.5 date_time_schema function definitions ... 59
11.5.1 acyclic_event_occurrence_relationship ... 59
11.5.2 acyclic_time_interval_relationship ... 60
11.5.3 duration_is_positive ... 62
11.5.4 duration_is_negative ... 62
11.5.5 leap_year ... 62
11.5.6 valid_calendar_date ... 63
11.5.7 valid_time ... 64

12 document_schema ... 64
12.1 General ... 64
12.2 Fundamental concepts and assumptions ... 64
12.3 document_schema type definitions ... 65
12.3.1 product_or_formation_or_definition ... 65
12.4 document_schema entity definitions ... 65
12.4.1 document ... 65
12.4.2 document_product_association ... 66
12.4.3 document_relationship ... 66
12.4.4 document_representation_type ... 67
12.4.5 document_type ... 67

© ISO 2019 – All rights reserved

12.4.6 document_usage_constraint ... 68
12.4.7 document_with_class ... 68
12.5 document_schema function definitions ... 69
12.5.1 acyclic_document_relationship ... 69

13 effectivity_schema ... 70
13.1 General ... 70
13.2 Fundamental concepts and assumptions ... 71
13.3 effectivity_schema type definitions ... 71
13.3.1 effs_description_attribute_select ... 71
13.3.2 effs_name_attribute_select ... 71
13.4 effectivity_schema entity definitions ... 71
13.4.1 condition ... 71
13.4.2 conditional_effectivity ... 72
13.4.3 conditional_configuration ... 72
13.4.4 dated_effectivity ... 72
13.4.5 effectivity ... 73
13.4.6 effectivity_relationship ... 74
13.4.7 lot_effectivity ... 75
13.4.8 serial_numbered_effectivity ... 75
13.4.9 time_interval_based_effectivity ... 75
13.5 effectivity_schema function definitions ... 76
13.5.1 acyclic_effectivity_relationship ... 76

14 experience_schema ... 77
14.1 General ... 77
14.2 Fundamental concepts and assumptions ... 77
14.3 experience_schema entity definitions ... 78
14.3.1 experience ... 78
14.3.2 experience_relationship ... 78
14.3.3 experience_type ... 79
14.3.4 experience_type_relationship ... 80
14.4 experience_schema function definitions ... 81
14.4.1 acyclic_experience_relationship ... 81
14.4.2 acyclic_experience_type_relationship ... 82

15 external_reference_schema ... 83
15.1 General ... 83
15.2 Fundamental concepts and assumptions ... 84
15.3 external_reference_schema type definitions ... 84
15.3.1 ers_description_attribute_select ... 84
15.3.2 message ... 84
15.3.3 source_item ... 84
15.4 external_reference_schema entity definitions ... 84
15.4.1 external_source ... 84
15.4.2 external_source_relationship ... 85
15.4.3 externally_defined_item ... 86
15.4.4 externally_defined_item_relationship ... 86
15.4.5 externally_defined_item_with_multiple_references ... 87
15.4.6 pre_defined_item ... 87
15.5 external_reference_schema function definitions ... 88
15.5.1 acyclic_external_source_relationship ... 88
15.5.2 acyclic_externally_defined_item_relationship ... 89

16 group_schema ... 90
16.1 General ... 90
16.2 Fundamental concepts and assumptions ... 91
16.3 group_schema type definitions ... 91
16.3.1 gs_id_attribute_select ... 91
16.4 group_schema entity definitions ... 91
16.4.1 group ... 91

 © ISO 2019 – All rights reserved

16.4.2 group_relationship ... 92
16.5 group_schema function definitions ... 93
16.5.1 acyclic_group_relationship ... 93

17 language_schema ... 94
17.1 General ... 94
17.2 Fundamental concepts and assumptions ... 94
17.3 language_schema entity definitions ... 94
17.3.1 language ... 94

18 location_schema ... 95
18.1 General ... 95
18.2 Fundamental concepts and assumptions ... 95
18.3 location_schema entity definitions ... 95
18.3.1 location ... 95
18.3.2 location_relationship ... 96
18.4 location_schema function definitions ... 96
18.4.1 acyclic_location_relationship ... 96

19 management_resources_schema ... 98
19.1 General ... 98
19.2 Fundamental concepts and assumptions ... 98
19.3 management_resources_schema type definitions ... 99
19.3.1 mrs_role_select ... 99
19.3.2 assignment_object_select ... 99
19.3.3 attribute_type ... 99
19.4 management_resources_schema entity definitions ... 99
19.4.1 action_assignment ... 99
19.4.2 action_method_assignment ... 100
19.4.3 action_method_role ... 100
19.4.4 action_request_assignment ... 101
19.4.5 approval_assignment ... 101
19.4.6 assignment_object_relationship ... 102
19.4.7 attribute_classification_assignment ... 103
19.4.8 attribute_value_assignment ... 103
19.4.9 attribute_value_role ... 104
19.4.10 certification_assignment ... 104
19.4.11 classification_assignment ... 105
19.4.12 classification_assignment_relationship ... 105
19.4.13 classification_role ... 106
19.4.14 contract_assignment ... 106
19.4.15 date_and_time_assignment ... 106
19.4.16 date_assignment ... 107
19.4.17 document_reference ... 107
19.4.18 document_usage_constraint_assignment ... 108
19.4.19 document_usage_role ... 109
19.4.20 effectivity_assignment ... 109
19.4.21 effectivity_context_assignment ... 110
19.4.22 effectivity_context_role ... 110
19.4.23 event_occurrence_assignment ... 110
19.4.24 event_occurrence_context_assignment ... 111
19.4.25 experience_assignment ... 111
19.4.26 experience_role ... 112
19.4.27 experience_type_assignment ... 112
19.4.28 experience_type_role ... 113
19.4.29 external_identification_assignment ... 113
19.4.30 external_identification_assignment_relationship ... 113
19.4.31 external_referent_assignment ... 114
19.4.32 group_assignment ... 115
19.4.33 identification_assignment ... 115
19.4.34 identification_assignment_relationship ... 116

© ISO 2019 – All rights reserved

19.4.35 identification_role 117
19.4.36 library_assignment 117
19.4.37 location_assignment 117
19.4.38 location_representation_assignment 118
19.4.39 location_representation_role 118
19.4.40 location_role 119
19.4.41 name_assignment 119
19.4.42 organization_assignment 120
19.4.43 organization_type_assignment 120
19.4.44 organization_type_role 120
19.4.45 organizational_project_assignment 121
19.4.46 organizational_project_role 121
19.4.47 person_and_organization_assignment 122
19.4.48 person_assignment 122
19.4.49 person_type_assignment 122
19.4.50 person_type_definition_assignment 123
19.4.51 person_type_definition_role 124
19.4.52 person_type_role 124
19.4.53 position_in_organization_assignment 124
19.4.54 position_in_organization_role 125
19.4.55 position_in_organization_type_assignment 125
19.4.56 position_in_organization_type_role 126
19.4.57 qualification_assignment 126
19.4.58 qualification_role 127
19.4.59 qualification_type_assignment 127
19.4.60 qualification_type_role 128
19.4.61 security_classification_assignment 128
19.4.62 time_assignment 129
19.4.63 time_interval_assignment 129
19.5 management_resources_schema function definitions 129
19.5.1 acyclic_assignment_object_relationship 129
19.5.2 acyclic_classification_assignment_relationship 131
19.5.3 acyclic_external_identification_assignment_relationship 132
19.5.4 acyclic_identification_assignment_relationship 133

20 measure_schema 135
20.1 General 135
20.2 Fundamental concepts and assumptions 135
20.3 measure_schema type definitions 135
20.3.1 s_name_attribute_select 135
20.3.2 absorbed_dose_measure 136
20.3.3 acceleration_measure 136
20.3.4 radioactivity_measure 136
20.3.5 amount_of_substance_measure 136
20.3.6 area_measure 136
20.3.7 celsius_temperature_measure 136
20.3.8 context_dependent_measure 136
20.3.9 count_measure 137
20.3.10 descriptive_measure 137
20.3.11 capacitance_measure 137
20.3.12 dose_equivalent_measure 137
20.3.13 electric_charge_measure 137
20.3.14 electric_current_measure 137
20.3.15 conductance_measure 137
20.3.16 electric_potential_measure 138
20.3.17 energy_measure 138
20.3.18 force_measure 138
20.3.19 frequency_measure 138
20.3.20 magnetic_flux_density_measure 138
20.3.21 illuminance_measure 138

 © ISO 2019 – All rights reserved

20.3.22 inductance_measure ... 138
20.3.23 length_measure ... 139
20.3.24 luminous_intensity_measure ... 139
20.3.25 luminous_flux_measure ... 139
20.3.26 mass_measure ... 139
20.3.27 measure_value ... 139
20.3.28 magnetic_flux_measure ... 140
20.3.29 non_negative_length_measure ... 140
20.3.30 numeric_measure ... 140
20.3.31 parameter_value ... 140
20.3.32 plane_angle_measure ... 141
20.3.33 positive_length_measure ... 141
20.3.34 positive_plane_angle_measure ... 141
20.3.35 positive_ratio_measure ... 141
20.3.36 power_measure ... 141
20.3.37 pressure_measure ... 141
20.3.38 ratio_measure ... 142
20.3.39 resistance_measure ... 142
20.3.40 si_prefix ... 142
20.3.41 si_unit_name ... 142
20.3.42 solid_angle_measure ... 143
20.3.43 thermodynamic_temperature_measure ... 143
20.3.44 time_measure ... 143
20.3.45 unit ... 143
20.3.46 velocity_measure ... 143
20.3.47 volume_measure ... 144
20.4 measure_schema entity definitions ... 144
20.4.1 absorbed_dose_measure_with_unit ... 144
20.4.2 absorbed_dose_unit ... 144
20.4.3 si_absorbed_dose_unit ... 144
20.4.4 acceleration_measure_with_unit ... 145
20.4.5 acceleration_unit ... 145
20.4.6 radioactivity_measure_with_unit ... 145
20.4.7 radioactivity_unit ... 146
20.4.8 si_radioactivity_unit ... 146
20.4.9 amount_of_substance_measure_with_unit ... 146
20.4.10 amount_of_substance_unit ... 146
20.4.11 area_measure_with_unit ... 147
20.4.12 area_unit ... 147
20.4.13 celsius_temperature_measure_with_unit ... 148
20.4.14 context_dependent_unit ... 148
20.4.15 conversion_based_unit ... 148
20.4.16 derived_unit ... 149
20.4.17 derived_unit_element ... 150
20.4.18 dimensional_exponents ... 150
20.4.19 capacitance_unit ... 151
20.4.20 capacitance_measure_with_unit ... 151
20.4.21 si_capacitance_unit ... 151
20.4.22 dose_equivalent_measure_with_unit ... 152
20.4.23 dose_equivalent_unit ... 152
20.4.24 si_dose_equivalent_unit ... 152
20.4.25 electric_charge_unit ... 152
20.4.26 electric_charge_measure_with_unit ... 153
20.4.27 si_electric_charge_unit ... 153
20.4.28 electric_current_measure_with_unit ... 153
20.4.29 electric_current_unit ... 154
20.4.30 conductance_unit ... 154
20.4.31 conductance_measure_with_unit ... 154
20.4.32 si_conductance_unit ... 155
20.4.33 electric_potential_unit ... 155

© ISO 2019 – All rights reserved

20.4.34 electric_potential_measure_with_unit 155
20.4.35 si_electric_potential_unit 156
20.4.36 energy_unit 156
20.4.37 energy_measure_with_unit 156
20.4.38 si_energy_unit 156
20.4.39 magnetic_flux_density_unit 157
20.4.40 magnetic_flux_density_measure_with_unit 157
20.4.41 si_magnetic_flux_density_unit 157
20.4.42 force_unit 158
20.4.43 force_measure_with_unit 158
20.4.44 si_force_unit 158
20.4.45 frequency_unit 159
20.4.46 frequency_measure_with_unit 159
20.4.47 si_frequency_unit 159
20.4.48 global_unit_assigned_context 159
20.4.49 illuminance_unit 160
20.4.50 illuminance_measure_with_unit 160
20.4.51 si_illuminance_unit 160
20.4.52 inductance_unit 161
20.4.53 inductance_measure_with_unit 161
20.4.54 si_inductance_unit 161
20.4.55 length_measure_with_unit 162
20.4.56 positive_length_measure_with_unit 162
20.4.57 length_unit 162
20.4.58 luminous_flux_unit 163
20.4.59 luminous_flux_measure_with_unit 163
20.4.60 luminous_intensity_measure_with_unit 163
20.4.61 luminous_intensity_unit 163
20.4.62 magnetic_flux_unit 164
20.4.63 magnetic_flux_measure_with_unit 164
20.4.64 si_magnetic_flux_unit 164
20.4.65 mass_measure_with_unit 165
20.4.66 mass_unit 165
20.4.67 measure_with_unit 165
20.4.68 named_unit 166
20.4.69 plane_angle_measure_with_unit 167
20.4.70 positive_plane_angle_measure_with_unit 167
20.4.71 plane_angle_unit 167
20.4.72 power_unit 168
20.4.73 power_measure_with_unit 168
20.4.74 si_power_unit 168
20.4.75 pressure_unit 168
20.4.76 pressure_measure_with_unit 169
20.4.77 si_pressure_unit 169
20.4.78 ratio_measure_with_unit 169
20.4.79 ratio_unit 170
20.4.80 resistance_unit 170
20.4.81 resistance_measure_with_unit 170
20.4.82 si_resistance_unit 171
20.4.83 si_unit 171
20.4.84 solid_angle_measure_with_unit 171
20.4.85 solid_angle_unit 172
20.4.86 thermodynamic_temperature_measure_with_unit 172
20.4.87 thermodynamic_temperature_unit 172
20.4.88 time_measure_with_unit 173
20.4.89 time_unit 173
20.4.90 velocity_measure_with_unit 173
20.4.91 velocity_unit 174
20.4.92 volume_measure_with_unit 174
20.4.93 volume_unit 174

 © ISO 2019 – All rights reserved

20.5 measure_schema function definitions ... 175
20.5.1 derive_dimensional_exponents ... 175
20.5.2 dimensions_for_si_unit ... 176
20.5.3 valid_units ... 177

21 person_organization_schema ... 180
21.1 General ... 180
21.2 Fundamental concepts and assumptions ... 181
21.3 person_organization_schema type definitions ... 181
21.3.1 person_organization_select ... 181
21.3.2 pos_description_attribute_select ... 181
21.3.3 pos_generic_attribute_select ... 181
21.3.4 pos_generic_attribute_list_select ... 181
21.3.5 pos_id_attribute_select ... 182
21.3.6 pos_name_attribute_select ... 182
21.4 person_organization_schema entity definitions ... 182
21.4.1 address ... 182
21.4.2 organization ... 183
21.4.3 organization_relationship ... 184
21.4.4 organization_role ... 185
21.4.5 organization_type ... 185
21.4.6 organization_type_relationship ... 186
21.4.7 organizational_address ... 187
21.4.8 organizational_project ... 187
21.4.9 organizational_project_relationship ... 188
21.4.10 person ... 188
21.4.11 person_and_organization ... 189
21.4.12 person_and_organization_role ... 190
21.4.13 person_role ... 191
21.4.14 person_type ... 192
21.4.15 person_type_definition ... 192
21.4.16 person_type_definition_formation ... 192
21.4.17 person_type_definition_relationship ... 193
21.4.18 personal_address ... 194
21.4.19 position_in_organization ... 194
21.4.20 position_in_organization_relationship ... 194
21.4.21 position_in_organization_type ... 195
21.5 person_organization_schema function definitions ... 196
21.5.1 acyclic_organization_relationship ... 196
21.5.2 acyclic_organization_type_relationship ... 197
21.5.3 acyclic_organizational_project_relationship ... 198
21.5.4 acyclic_person_type_definition_relationship ... 200
21.5.5 acyclic_position_in_organization_relationship ... 201

22 product_definition_schema ... 202
22.1 General ... 202
22.2 Fundamental concepts and assumptions ... 203
22.3 product_definition_schema type definitions ... 204
22.3.1 product_definition_or_reference ... 204
22.3.2 pds_id_attribute_select ... 204
22.3.3 pds_name_attribute_select ... 204
22.3.4 source ... 205
22.4 product_definition_schema entity definitions ... 205
22.4.1 alternative_solution_relationship ... 205
22.4.2 characterized_product ... 205
22.4.3 generic_product_definition_reference ... 205
22.4.4 product ... 206
22.4.5 product_category ... 207
22.4.6 product_category_relationship ... 208
22.4.7 product_definition ... 209

© ISO 2019 – All rights reserved

22.4.8 product_definition_context_association 210
22.4.9 product_definition_context_role 211
22.4.10 product_definition_effectivity 211
22.4.11 product_definition_formation 211
22.4.12 final_solution 212
22.4.13 product_definition_formation_relationship 213
22.4.14 product_definition_formation_with_specified_source 214
22.4.15 product_definition_relationship 214
22.4.16 product_definition_relationship_relationship 215
22.4.17 product_definition_reference 216
22.4.18 product_definition_reference_with_local_representation 216
22.4.19 product_definition_substitute 217
22.4.20 product_definition_with_associated_documents 218
22.4.21 product_related_product_category 218
22.4.22 product_relationship 218
22.5 product_definition_schema function definitions 219
22.5.1 acyclic_alternative_solution_relationship 219
22.5.2 acyclic_product_category_relationship 220
22.5.3 acyclic_product_definition_formation_relationship 222
22.5.4 acyclic_product_definition_relationship 223
22.5.5 acyclic_product_relationship 224
22.5.6 categories_of_product 226
22.5.7 get_product_definitions 226
23 product_property_definition_schema 227
23.1 General 227
23.2 Fundamental concepts and assumptions 228
23.3 product_property_definition_schema type definitions 229
23.3.1 characterized_definition 229
23.3.2 characterized_item 229
23.3.3 ppds_id_attribute_select 229
23.3.4 characterized_product_definition 229
23.3.5 derived_property_select 230
23.3.6 shape_aspect_or_characterized_object 230
23.3.7 shape_definition 230
23.3.8 internal_or_reflected_shape_aspect 230
23.3.9 multi_or_next_assembly_usage_occurrence 231
23.4 product_property_definition_schema entity definitions 231
23.4.1 characterized_object 231
23.4.2 characterized_object_relationship 231
23.4.3 component_path_shape_aspect 232
23.4.4 constituent_shape_aspect 232
23.4.5 general_property 233
23.4.6 general_property_association 233
23.4.7 general_property_relationship 234
23.4.8 product_definition_shape 235
23.4.9 property_definition 235
23.4.10 shape_aspect 236
23.4.11 shape_aspect_occurrence 237
23.4.12 shape_aspect_relationship 238
23.5 product_property_definition_schema function definitions 239
23.5.1 acyclic_characterized_object_relationship 239
23.5.2 acyclic_general_property_relationship 241
23.5.3 acyclic_shape_aspect_occurrence 242
23.5.4 acyclic_shape_aspect_relationship 243
23.5.5 get_shape_aspects 244
23.5.6 get_root_shape_aspect 245
24 product_property_representation_schema 246
24.1 General 246

 © ISO 2019 – All rights reserved

24.2 Fundamental concepts and assumptions 247
24.3 product_property_representation_schema type definitions 247
24.3.1 pprs_description_attribute_select 247
24.3.2 pprs_name_attribute_select 247
24.3.3 chained_representation_link 247
24.3.4 represented_definition 247
24.3.5 item_identified_representation_usage_definition 248
24.3.6 item_identified_representation_usage_select 248
24.4 product_property_representation_schema entity definitions 248
24.4.1 chain_based_item_identified_representation_usage 248
24.4.2 characterized_chain_based_item_within_representation 249
24.4.3 characterized_item_within_representation 250
24.4.4 context_dependent_shape_representation 250
24.4.5 item_identified_representation_usage 251
24.4.6 property_definition_representation 253
24.4.7 shape_definition_representation 254
24.4.8 shape_representation 254
24.4.9 shape_representation_reference 254
24.4.10 shape_representation_relationship 254
24.4.11 specified_occurrence_context_dependent_shape_representation 255
24.5 product_property_representation_schema function definitions 255
24.5.1 get_directed_link 255
24.5.2 relatives_of_product_definitions 258
24.5.3 relatives_of_shape_representations 259
24.5.4 get_property_definition_representations 259

25 qualifications_schema 260
25.1 General 260
25.2 Fundamental concepts and assumptions 261
25.3 qualifications_schema entity definitions 261
25.3.1 qualification 261
25.3.2 qualification_relationship 261
25.3.3 qualification_type 262
25.3.4 qualification_type_relationship 262
25.4 qualifications_schema function definitions 263
25.4.1 acyclic_qualification_relationship 263
25.4.2 acyclic_qualification_type_relationship 265

26 security_classification_schema 266
26.1 General 266
26.2 Fundamental concepts and assumptions 266
26.3 security_classification_schema entity definitions 266
26.3.1 security_classification 266
26.3.2 security_classification_level 267

27 support_resource_schema 267
27.1 General 267
27.2 Fundamental concepts and assumptions 267
27.3 support_resource_schema type definitions 268
27.3.1 identifier 268
27.3.2 label 268
27.3.3 text 268
27.4 support_resource_schema function definitions 268
27.4.1 bag_to_set 268
27.4.2 type_check_function 269

© ISO 2019 – All rights reserved

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

Part 41:
Integrated generic resource: Fundamentals of product description and support

1 Scope

Scope.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org

3.1
Test term
Test definition

4 Twenty-four schemas to follow!

5 action_schema

5.1 General

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.

EXPRESS specification:

```
SCHEMA action_schema '{iso standard 10303 part(41) version(8) object(1) action_
schema(1)}';

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   description_attribute_select,
   get_description_value,
   get_id_value,
   get_name_value,
```

© ISO 2019 – All rights reserved

```
   get_role,
   id_attribute,
   id_attribute_select,
   name_attribute,
   name_attribute_select,
   object_role,
   role_select,
   role_association);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

5.2 Fundamental concepts and assumptions

Action information can be attached to any aspect of product data.

5.3 action_schema type definitions

5.3.1 as_description_attribute_select

EXPRESS specification:

```
TYPE as_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (action_request_solution);
END_TYPE;
```

5.3.2 as_id_attribute_select

EXPRESS specification:

```
TYPE as_id_attribute_select = SELECT BASED_ON id_attribute_select WITH
  (action);
END_TYPE;
```

5.3.3 as_name_attribute_select

EXPRESS specification:

```
TYPE as_name_attribute_select = SELECT BASED_ON name_attribute_select WITH
  (action_request_solution);
END_TYPE;
```

5.3.4 as_role_select

EXPRESS specification:

```
TYPE as_role_select = SELECT BASED_ON role_select WITH
  (directed_action_assignment);
END_TYPE;
```

5.3.5 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,
```

 © ISO 2019 – All rights reserved

```
   action_method);
END_TYPE;
```

5.4 action_schema entity definitions

5.4.1 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;
```

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.

Attribute definitions:

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

NOTE 5 The id_attribute data type is defined in basic_attribute_schema of this part of ISO 10303.

© ISO 2019 – All rights reserved

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

5.4.2 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;
```

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.

5.4.3 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;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

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**.

5.4.4 action_method_relationship

An **action_method_relationship** is a relationship between 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;
```

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.

5.4.5 action_relationship

An **action_relationship** is a relationship between two instances of the entity data type action and provides an identification and description of this relationship.

© ISO 2019 – All rights reserved

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;
```

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.

5.4.6 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;
```

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**.

 © ISO 2019 – All rights reserved

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.

Attribute definitions:

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 basic_attribute_schema 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 basic_attribute_schema 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.

5.4.7 action_request_status

An **action_request_status** is the association of a status with a versioned_action_request.

EXPRESS specification:

```
ENTITY action_request_status;
  status : label;
  assigned_request : versioned_action_request;
END_ENTITY;
```

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.

5.4.8 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;
```

© ISO 2019 – All rights reserved

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.

5.4.9 action_resource_relationship

An **action_resource_relationship** is a relationship between 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;
```

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.

5.4.10 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 a process operation.

EXPRESS specification:

```
ENTITY action_resource_type;
```


© ISO 2019 – All rights reserved

```
  name : label;
END_ENTITY;
```

Attribute definitions:

name the label by which the **action_resource_type** is known.

5.4.11 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;
```

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.

5.4.12 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;
```

Attribute definitions:

directive the action_directive that governs the **directed_action**.

5.4.13 directed_action_assignment

A **directed_action_assignment** is an association of a directed_action with product data.

EXPRESS specification:

```
ENTITY directed_action_assignment
```

© ISO 2019 – All rights reserved

```
  ABSTRACT SUPERTYPE;
  assigned_directed_action : directed_action;
DERIVE
  role : object_role := get_role(SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'BASIC_ATTRIBUTE_SCHEMA.' + 'ROLE_ASSOCIATION.ITEM_WITH_
ROLE')) <= 1;
END_ENTITY;
```

Attribute definitions:

assigned_directed_action the instance of the **directed_action** entity data that is to be associated with product data.

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

Attribute definitions:

WR1 Each **directed_action_assignment** shall be **item_with_role** in at most one role_association.

5.4.14 executed_action

An **executed_action** is a type of action that is 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;
```

5.4.15 versioned_action_request

A **versioned_action_request** is a specification of a desired result.

NOTE 1 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 : OPTIONAL label;
  purpose : text;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **versioned_action_request**.

 © ISO 2019 – All rights reserved

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.

version the identification of the version of the **versioned_action_request**. The value of the attribute need not be specified.

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.

5.4.16 versioned_action_request_relationship

A **versioned_action_request_relationship** is a relationship between two versioned_action_request objects.

NOTE 1 Two versioned_action_request objects may be related if they address similar problems.

NOTE 2 A versioned_action_request may be a version of a work request. It might be related to a different version of the work request using a **versioned_action_request_relationship**.

EXPRESS specification:

```
ENTITY versioned_action_request_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_versioned_action_request : versioned_action_request;
  related_versioned_action_request : versioned_action_request;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **versioned_action_request_relationship**.

name the label by which the **versioned_action_request_relationship** is known.

description the text that characterizes the **versioned_action_request_relationship**. The value of this attribute need not be specified.

relating_versioned_action_request one of the instances of versioned_action_request 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_versioned_action_request the other instance of versioned_action_request 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.

© ISO 2019 – All rights reserved

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.

5.4.17 action_directive_relationship

An **action_directive_relationship** is a relationship between two action_directive objects.

EXPRESS specification:

```
ENTITY action_directive_relationship;
  name : label;
  description : OPTIONAL STRING;
  relating : action_directive;
  related : action_directive;
  relation_type : STRING;
WHERE
  WR1: acyclic_action_directive_relationship(SELF, [related], 'ACTION_SCHEMA.ACTION_
DIRECTIVE_RELATIONSHIP');
END_ENTITY;
```

Attribute definitions:

name shall be unique within a population of **action_directive_relationship**.

description the text or the set of texts that provide further information about the **action_directive_relationship**. The value of this attribute need not be specified.

relating one object of action_directive that is part of the relationship.

related the other object of action_directive that is part of the relationship.

relation_type the meaning of the relationship.

Attribute definitions:

WR1 A **action_directive_relationship** shall not participate in its own definition.

5.5 action_schema function definitions

5.5.1 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 an 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.relating_method IN relatives THEN
```

 © ISO 2019 – All rights reserved

```
    RETURN (FALSE);
  END_IF;
  x := QUERY(am <* bag_to_set(USEDIN(relation.relating_method, ('ACTION_SCHEMA.' +
'ACTION_METHOD_RELATIONSHIP.') + 'RELATED')) | specific_relation IN TYPEOF(am));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_action_method_relationship(x[i], relatives + relation.relating_method,
specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

5.5.1.1 Parameters

5.5.1.1.1 relation

(input) the candidate action_method_relationship to be checked.

EXPRESS specification:

```
relation : action_method_relationship
```

5.5.1.1.2 relatives

(input) the set of instances of the entity data type action_method that the function is searching for in the relating_method parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF action_method
```

5.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the action_method_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

5.5.1.2 Variables

5.5.1.2.1 x

EXPRESS specification:

```
x : SET OF action_method_relationship;
```

5.5.1.3 Statements

5.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_action_method_relationship(x[i], relatives + relation.relating_method,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

© ISO 2019 – All rights reserved

5.5.2 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 an 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;
```

5.5.2.1 Parameters

5.5.2.1.1 relation

(input) the candidate action_relationship to be checked.

EXPRESS specification:

```
relation : action_relationship
```

5.5.2.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF action
```

5.5.2.1.3 specific_relation

(input) the fully qualified name of a subtype of the action_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

 © ISO 2019 – All rights reserved

5.5.2.2 Variables

5.5.2.2.1 x

EXPRESS specification:

```
x : SET OF action_relationship;
```

5.5.2.3 Statements

5.5.2.3.1 i

EXPRESS specification:

```
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;
```

5.5.3 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 an 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;
```

5.5.3.1 Parameters

5.5.3.1.1 relation

(input) the candidate action_resource_relationship to be checked.

EXPRESS specification:

© ISO 2019 – All rights reserved

```
relation : action_resource_relationship
```

5.5.3.1.2 relatives

(input) the set of instances of the entity data type action_resource that the function is searching for in the relating_resource parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF action_resource
```

5.5.3.1.3 specific_relation

(input) the fully qualified name of a subtype of the action_resource_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

5.5.3.2 Variables

5.5.3.2.1 x

EXPRESS specification:

```
x : SET OF action_resource_relationship;
```

5.5.3.3 Statements

5.5.3.3.1 i

EXPRESS specification:

```
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;
```

5.5.4 acyclic_versioned_action_request_relationship

The **acyclic_versioned_action_request_relationship** function determines whether the graph of instances of the entity data type versioned_action_request that contains relation as one of its links contains a cycle. This function may be used to evaluate either an 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 versioned_action_request_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_versioned_action_request_relationship(relation : versioned_action_
request_relationship;
                                                       relatives : SET OF versioned_
action_request;
                                                       specific_relation : STRING) :
BOOLEAN;
  LOCAL
    x : SET OF versioned_action_request_relationship;
  END_LOCAL;
```

 © ISO 2019 – All rights reserved

```
  IF relation.relating_versioned_action_request IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(varr <* bag_to_set(USEDIN(relation.relating_versioned_action_request,
('ACTION_SCHEMA.' + 'VERSIONED_ACTION_REQUEST_RELATIONSHIP.') + 'RELATED_VERSIONED_ACTION_
REQUEST')) | specific_relation IN TYPEOF(varr));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_versioned_action_request_relationship(x[i], relatives + relation.
relating_versioned_action_request, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

5.5.4.1 Parameters

5.5.4.1.1 relation

(input) the candidate versioned_action_request_relationship to be checked.

EXPRESS specification:

```
relation : versioned_action_request_relationship
```

5.5.4.1.2 relatives

(input) the set of instances of the entity data type versioned_action_request that the function is searching for in the relating_versioned_action_request parameter of the relation argument.

EXPRESS specification:

```
relatives : SET OF versioned_action_request
```

5.5.4.1.3 specific_relation

(input) the fully qualified name of a subtype of the versioned_action_request_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

5.5.4.2 Variables

5.5.4.2.1 x

EXPRESS specification:

```
x : SET OF versioned_action_request_relationship;
```

5.5.4.3 Statements

5.5.4.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_versioned_action_request_relationship(x[i], relatives + relation.
relating_versioned_action_request, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

© ISO 2019 – All rights reserved

5.5.5 acyclic_action_directive_relationship

The **acyclic_action_directive_relationship** function determines whether the graph of instance of the entity data type action_directive that contains relation as one of its links contains a cycle. this function may be used to evaluate either a action_directive_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_action_directive_relationship(relation : action_directive_relationship;
                                               relatives : SET [1:?] OF action_directive;
                                               specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF action_directive_relationship;
  END_LOCAL;
  IF relation.relating IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(adr <* bag_to_set(USEDIN(relation.relating, ('ACTION_SCHEMA.' + 'ACTION_
DIRECTIVE_RELATIONSHIP.') + 'RELATED')) | specific_relation IN TYPEOF(adr));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_action_directive_relationship(x[i], relatives + relation.relating,
specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

5.5.5.1 Parameters

5.5.5.1.1 relation

(input) the candidate action_directive_relationship to be checked.

EXPRESS specification:

```
relation : action_directive_relationship
```

5.5.5.1.2 relatives

(input) the set of instances of the entity data type action_directive for which the function is searching in the relating parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF action_directive
```

5.5.5.1.3 specific_relation

(input) the fully qualified name of a subtype of the action_directive_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

5.5.5.2 Variables

5.5.5.2.1 x

EXPRESS specification:

 © ISO 2019 – All rights reserved

```
x : SET OF action_directive_relationship;
```

5.5.5.3 Statements

5.5.5.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_action_directive_relationship(x[i], relatives + relation.relating,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

6 application_context_schema

6.1 General

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.

EXPRESS specification:

```
SCHEMA application_context_schema '{iso standard 10303 part(41) version(8) object(1)
application_context_schema(2)}';

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   description_attribute_select,
   get_description_value,
   get_id_value,
   id_attribute,
   id_attribute_select);

REFERENCE FROM date_time_schema
  (year_number);

REFERENCE FROM language_schema
  (language);

REFERENCE FROM management_resources_schema
  (identification_assignment);

REFERENCE FROM person_organization_schema
  (organization);

REFERENCE FROM support_resource_schema
  (identifier,
   label,
   text);
```

6.2 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.

© ISO 2019 – All rights reserved

6.3 application_context_schema type definitions

6.3.1 acs_description_attribute_select

EXPRESS specification:

```
TYPE acs_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (application_context);
END_TYPE;
```

6.3.2 acs_id_attribute_select

EXPRESS specification:

```
TYPE acs_id_attribute_select = SELECT BASED_ON id_attribute_select WITH
  (application_context);
END_TYPE;
```

6.3.3 exchange_identification_context_select

The **exchange_identification_context_select** type allows for the designation of the data types identification_assignment and organization .

EXPRESS specification:

```
TYPE exchange_identification_context_select = SELECT
  (identification_assignment,
   organization);
END_TYPE;
```

6.4 application_context_schema entity definitions

6.4.1 application_context

An **application_context**, as defined in ISO 10303-1, 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;
```

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.

 © ISO 2019 – All rights reserved

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**.

Attribute definitions:

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 basic_attribute_schema 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 basic_attribute_schema 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.

6.4.2 application_context_element

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

An **application_context_element** may be either a library_context or a product_concept_context, or a product_context, or a product_definition_context.

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;
```

Attribute definitions:

© ISO 2019 – All rights reserved

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

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

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

6.4.3 application_context_relationship

An **application_context_relationship** is a relationship between 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;
```

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.

© ISO 2019 – All rights reserved

6.4.4 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;
```

Attribute definitions:

status the label that provides a user interpretable designation for the level of completion of the application protocol.

EXAMPLE "IS", "DIS", "TS" or "CD" are examples of application protocol states.

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.

6.4.5 library_context

A **library_context** is a type of application_context_element that defines a context for elements of a library.

The inherited attribute name 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.

EXPRESS specification:

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

Attribute definitions:

library_reference the label that identifies the library.

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

6.4.6 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.

© ISO 2019 – All rights reserved

EXPRESS specification:

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

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_type*s.

6.4.7 product_context

A **product_context** is a type of application_context_element that represents life cycle independent information about a product. This information describes the discipline in which data about the product are created.

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'.

The inherited attribute name 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.

EXPRESS specification:

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

Attribute definitions:

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', or 'architecture' are examples of discipline types.

6.4.8 product_definition_context

A **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 is created or used.

 © ISO 2019 – All rights reserved

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'.

The inherited attribute name 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'.

EXPRESS specification:

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

Attribute definitions:

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

6.4.9 dependent_product_definition_exchange_context

A **dependent_product_definition_exchange_context** is a type of product_definition_context and is an abstract entity that describes a specific context for exchange.

EXPRESS specification:

```
ENTITY dependent_product_definition_exchange_context
  ABSTRACT SUPERTYPE
  SUBTYPE OF (product_definition_context);
WHERE
  WR1: SIZEOF(TYPEOF(SELF) * ['APPLICATION_CONTEXT_SCHEMA.INDENTIFICATION_DEPENDENT_
PRODUCT_DEFINITION_EXCHANGE_CONTEXT', 'APPLICATION_CONTEXT_SCHEMA.LANGUAGE_DEPENDENT_
PRODUCT_DEFINITION_EXCHANGE_CONTEXT']) > 0;
END_ENTITY;
```

Attribute definitions:

WR1 at least one of the entities identification_dependent_product_definition_exchange_context or language_dependent_product_definition_exchange_context shall be specified.

© ISO 2019 – All rights reserved

6.4.10 identification_dependent_product_definition_exchange_context

An **identification_dependent_product_definition_exchange_context** is a type of dependent_product_definition_exchange_context that establishes the context of the product_definition exchange.

EXPRESS specification:

```
ENTITY identification_dependent_product_definition_exchange_context
  SUBTYPE OF (dependent_product_definition_exchange_context);
  identification_context : exchange_identification_context_select;
END_ENTITY;
```

Attribute definitions:

identification_context specifies the context of the exchange to be either identification_assignment or organization .

6.4.11 language_dependent_product_definition_exchange_context

A **language_dependent_product_definition_exchange_context** is a type of dependent_product_definition_exchange_context that establishes the specified language of the product_definition exchange context.

EXPRESS specification:

```
ENTITY language_dependent_product_definition_exchange_context
  SUBTYPE OF (dependent_product_definition_exchange_context);
  language : language;
END_ENTITY;
```

Attribute definitions:

language specifies the default language of the exchange context.

7 approval_schema

7.1 General

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.

EXPRESS specification:

```
SCHEMA approval_schema '{iso standard 10303 part(41) version(7) object(1) approval_
schema(3)}';

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   description_attribute_select,
   get_description_value,
   get_role,
   object_role,
   role_association,
   role_select);

REFERENCE FROM date_time_schema
  (date_time_select);
```


© ISO 2019 – All rights reserved

```
REFERENCE FROM person_organization_schema
  (person_organization_select);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   label,
   text);
```

7.2 Fundamental concepts and assumptions

Approval information can be attached to any aspect of product data.

7.3 approval_schema type definitions

7.3.1 apps_description_attribute_select

EXPRESS specification:

```
TYPE apps_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (approval_role);
END_TYPE;
```

7.3.2 apps_role_select

EXPRESS specification:

```
TYPE apps_role_select = SELECT BASED_ON role_select WITH
  (approval_date_time);
END_TYPE;
```

7.4 approval_schema entity definitions

7.4.1 approval

An **approval** is a confirmation of the product data for its intended use. An **approval** is an incompletely modelled concept and requires population of related entity data types to form a complete approval.

NOTE 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;
```

Attribute definitions:

status the type or level of **approval** in terms of its usage.

EXAMPLE One possible status is 'released for production'; this explicitly identifies the approved usage. Another possible status is 'preliminary design completed'; this only implies the approved usage that will depend upon company-specific procedures.

level the label that provides a user interpretable designation for the level of completion of the approval.

© ISO 2019 – All rights reserved

7.4.2 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;
```

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.

Attribute definitions:

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.

7.4.3 approval_person_organization

An **approval_person_organization** is an association between an approval and a given person, organization, or person in the context of an organization.

EXPRESS specification:

```
ENTITY approval_person_organization;
  person_organization : person_organization_select;
  authorized_approval : approval;
  role : approval_role;
END_ENTITY;
```

Attribute definitions:

person_organization the person or organization playing the given role.

authorized_approval the approval that is affected by the person or organization.

role the approval_role that specifies the purpose of the association of the approval with a person or organization.

© ISO 2019 – All rights reserved

7.4.4 approval_relationship

An **approval_relationship** is a relationship between 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;
```

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.

7.4.5 approval_role

An **approval_role** is a definition of a function. An **approval_role** is an incompletely modelled concept and requires population of related entity data types to form a complete confirmation.

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;
```

Attribute definitions:

role the label by which the **approval_role** is known.

© ISO 2019 – All rights reserved

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.

Attribute definitions:

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 basic_attribute_schema 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.

7.4.6 approval_status

An **approval_status** specifies the state of an approval.

EXAMPLE 'Approved' and 'disapproved' are examples of **approval_status**.

EXPRESS specification:

```
ENTITY approval_status;
  name : label;
END_ENTITY;
```

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.

7.5 approval_schema function definitions

7.5.1 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
```

 © ISO 2019 – All rights reserved

```
    x : SET OF approval_relationship;
  END_LOCAL;
  IF relation.relating_approval IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(app <* bag_to_set(USEDIN(relation.relating_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.relating_approval,
specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

7.5.1.1 Parameters

7.5.1.1.1 relation

(input) the candidate approval_relationship to be checked.

EXPRESS specification:

```
relation : approval_relationship
```

7.5.1.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF approval
```

7.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the approval_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

7.5.1.2 Variables

7.5.1.2.1 x

EXPRESS specification:

```
x : SET OF approval_relationship;
```

7.5.1.3 Statements

7.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_approval_relationship(x[i], relatives + relation.relating_approval,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

© ISO 2019 – All rights reserved

8 basic_attribute_schema

8.1 General

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.

EXPRESS specification:

```
SCHEMA basic_attribute_schema '{iso standard 10303 part(41) version(9) object(1) basic_
attribute_schema(4)}';

REFERENCE FROM support_resource_schema
  (identifier,
   label,
   text);
```

8.2 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.

 © ISO 2019 – All rights reserved

8.3 basic_attribute_schema type definitions

8.3.1 description_attribute_select

A **description_attribute_select** specifies those objects that may be assigned a descriptive text.

EXPRESS specification:

```
TYPE description_attribute_select = EXTENSIBLE GENERIC_ENTITY SELECT;
END_TYPE;
```

8.3.2 generic_attribute_list_select

A **generic_attribute_list_select** specifies those objects that may be assigned a generic attribute list.

EXPRESS specification:

```
TYPE generic_attribute_list_select = EXTENSIBLE GENERIC_ENTITY SELECT;
END_TYPE;
```

8.3.3 generic_attribute_select

A **generic_attribute_select** specifies those objects that may be assigned a generic attribute.

EXPRESS specification:

```
TYPE generic_attribute_select = EXTENSIBLE GENERIC_ENTITY SELECT;
END_TYPE;
```

8.3.4 id_attribute_select

An **id_attribute_select** specifies those objects that may be assigned an identifier.

EXPRESS specification:

```
TYPE id_attribute_select = EXTENSIBLE GENERIC_ENTITY SELECT;
END_TYPE;
```

8.3.5 name_attribute_select

A **name_attribute_select** specifies those objects that may be assigned a name.

EXPRESS specification:

```
TYPE name_attribute_select = EXTENSIBLE GENERIC_ENTITY SELECT;
END_TYPE;
```

8.3.6 role_select

A **role_select** specifies those objects that may be assigned a role.

EXPRESS specification:

```
TYPE role_select = EXTENSIBLE GENERIC_ENTITY SELECT;
END_TYPE;
```

8.4 basic_attribute_schema entity definitions

8.4.1 aggregate_id_attribute

An **aggregate_id_attribute** is used for the case where there is more than one product data element for a specific identifier.

© ISO 2019 – All rights reserved

NOTE 1 This entity provides the capability to add an aggregate 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 aggregate_id_attribute;
  attribute_value : identifier;
  identified_item : SET [1:?] OF id_attribute_select;
END_ENTITY;
```

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 be default, in an agreement of common understanding between partners sharing this information.

identified_item the collection of items that are identified.

8.4.2 description_attribute

A **description_attribute** is 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;
```

Attribute definitions:

attribute_value the text that characterizes the **described_item**.

described_item the item that is described.

8.4.3 generic_attribute

A **generic_attribute** is the declaration of a named attribute as a text string to enable it to be associated with product data. This capability shall not be used to replace more specific assignment capabilities if both are available to a specified target.

NOTE This entity provides the capability to define a generic attribute associated with product data using a method that is upwardly compatible with ISO 10303-41:1994, ISO 10303-43:1994, and ISO 10303-44:1994.

EXAMPLE A person entity data type instance is included in both generic_attribute_select and name_attribute_select. The name attribute of person is provided for the name value.

EXPRESS specification:

```
ENTITY generic_attribute;
  attribute_name : label;
  attribute_value : text;
END_ENTITY;
```

 © ISO 2019 – All rights reserved

Attribute definitions:

attribute_name the label used to name the **generic_attribute**.

attribute_value the text that characterizes the **generic_attribute**.

8.4.4 generic_attribute_assignment

A **generic_attribute_assignment** is the assignment of a generic attribute to element of product data.

NOTE This entity provides the capability to add a generic attribute to product data using a method that is upwardly compatible with ISO 10303-41:1994 and ISO 10303-43:1994.

EXPRESS specification:

```
ENTITY generic_attribute_assignment;
  attribute : generic_attribute;
  item_with_attribute : generic_attribute_select;
END_ENTITY;
```

Attribute definitions:

attribute the generic_attribute to be associated with the product data.

item_with_attribute the item to which the attribute is assigned.

8.4.5 generic_attribute_list_assignment

A **generic_attribute_list_assignment** is the assignment of a named list of generic attributes to element of product data.

NOTE This entity data type provides a capability to add a named list of generic attributes to certain product data entity data types using a method that is upwardly compatible with ISO 10303-41:1994, ISO 10303-43:1994, and ISO 10303-44:1994.

EXPRESS specification:

```
ENTITY generic_attribute_list_assignment;
  attribute_name : label;
  attributes : LIST [1:?] OF generic_attribute;
  item_with_attribute : generic_attribute_list_select;
END_ENTITY;
```

Attribute definitions:

attribute_name the label used to name the list of generic attributes.

attributes the list of generic_attributes that are assigned to the element of product data.

item_with_attribute the item to which the attributes are assigned.

8.4.6 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:

© ISO 2019 – All rights reserved

```
ENTITY id_attribute;
  attribute_value : identifier;
  identified_item : id_attribute_select;
END_ENTITY;
```

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 be default, in an agreement of common understanding between partners sharing this information.

identified_item the item that is identified.

8.4.7 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;
```

Attribute definitions:

attribute_value the label by which the **named_item** is known.

named_item the item to which the name is applied.

8.4.8 object_role

An **object_role** is a specification of 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;
```

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.

 © ISO 2019 – All rights reserved

8.4.9 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;
```

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.

8.5 basic_attribute_schema function definitions

8.5.1 get_description_value

The **get_description_value** function 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;
```

8.5.1.1 Parameters

8.5.1.1.1 obj

(input) the object for which a text is determined. This is an input to the function.

EXPRESS specification:

```
obj : description_attribute_select
```

8.5.1.2 Variables

8.5.1.2.1 description_bag

EXPRESS specification:

© ISO 2019 – All rights reserved

```
description_bag : BAG OF description_attribute := USEDIN(obj, ('BASIC_ATTRIBUTE_SCHEMA.'
+ 'DESCRIPTION_ATTRIBUTE.') + 'DESCRIBED_ITEM');
```

8.5.2 get_id_value

The **get_id_value** function 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;
```

8.5.2.1 Parameters

8.5.2.1.1 obj

(input) the object for which an identifier is determined. This is an input to the function.

EXPRESS specification:

```
obj : id_attribute_select
```

8.5.2.2 Variables

8.5.2.2.1 id_bag

EXPRESS specification:

```
id_bag : BAG OF id_attribute := USEDIN(obj, ('BASIC_ATTRIBUTE_SCHEMA.' + 'ID_ATTRIBUTE.')
+ 'IDENTIFIED_ITEM');
```

8.5.3 get_aggregate_id_value

The **get_aggregate_id_value** function accepts an object to which an identifier may be assigned and returns this identifier if present.

If an aggregate_id_attribute references the object, the function will return the value of the attribute_value attribute of this aggregate_id_attribute. Otherwise, the indeterminate value will be returned.

EXPRESS specification:

```
FUNCTION get_aggregate_id_value(obj : id_attribute_select) : identifier;
  LOCAL
    aggregate_id_bag : BAG OF aggregate_id_attribute := USEDIN(obj, ('BASIC_ATTRIBUTE_
SCHEMA.' + 'AGGREGATE_ID_ATTRIBUTE.') + 'IDENTIFIED_ITEM');
  END_LOCAL;
  IF SIZEOF(aggregate_id_bag) = 1 THEN
    RETURN (aggregate_id_bag[1].attribute_value);
  ELSE
    RETURN (?);
  END_IF;
```


© ISO 2019 – All rights reserved

```
END_FUNCTION;
```

8.5.3.1 Parameters

8.5.3.1.1 obj

(input) the object for which an identifier is determined. This is an input to the function.

EXPRESS specification:

```
obj : id_attribute_select
```

8.5.3.2 Variables

8.5.3.2.1 aggregate_id_bag

EXPRESS specification:

```
aggregate_id_bag : BAG OF aggregate_id_attribute := USEDIN(obj, ('BASIC_ATTRIBUTE_SCHEMA.
' + 'AGGREGATE_ID_ATTRIBUTE.') + 'IDENTIFIED_ITEM');
```

8.5.4 get_name_value

The **get_name_value** function 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;
```

8.5.4.1 Parameters

8.5.4.1.1 obj

the object for which a label is determined. This is an input to the function.

EXPRESS specification:

```
obj : name_attribute_select
```

8.5.4.2 Variables

8.5.4.2.1 name_bag

EXPRESS specification:

```
name_bag : BAG OF name_attribute := USEDIN(obj, ('BASIC_ATTRIBUTE_SCHEMA.' + 'NAME_
ATTRIBUTE.') + 'NAMED_ITEM');
```

© ISO 2019 – All rights reserved

8.5.5 get_role

The **get_role** function 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;
```

8.5.5.1 Parameters

8.5.5.1.1 obj

(input) the object for which an object_role is determined. This is an input to the function.

EXPRESS specification:

```
obj : role_select
```

8.5.5.2 Variables

8.5.5.2.1 role_bag

EXPRESS specification:

```
role_bag : BAG OF role_association := USEDIN(obj, ('BASIC_ATTRIBUTE_SCHEMA.' + 'ROLE_
ASSOCIATION.') + 'ITEM_WITH_ROLE');
```

9 certification_schema

9.1 General

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.

EXPRESS specification:

```
SCHEMA certification_schema '{iso standard 10303 part(41) version(3) object(1)
certification_schema(5)}';

REFERENCE FROM support_resource_schema
  (label,
   text);
```


© ISO 2019 – All rights reserved

9.2 Fundamental concepts and assumptions

Certification information can be attached to any aspect of product data.

9.3 certification_schema entity definitions

9.3.1 certification

A **certification** is an assertion of facts. A **certification** is an incompletely modeled concept and requires population of related entity data types to form a complete certification.

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;
```

Attribute definitions:

name the label by which the **certification** is known.

purpose an informal description of the reason 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.

9.3.2 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;
```

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 A certification type in an example supply chain is ‘supplier’.

© ISO 2019 – All rights reserved

10 contract_schema

10.1 General

The subject of the **contract_schema** is the description of contract agreements.

EXPRESS specification:

```
SCHEMA contract_schema '{iso standard 10303 part(41) version(3) object(1) contract_
schema(6)}';

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

10.2 Fundamental concepts and assumptions

Contract information may be attached to any aspect of a product data.

10.3 contract_schema entity definitions

10.3.1 contract

A **contract** is a binding agreement. A **contract** is an incompletely modeled concept and requires population of related entity data types to form a complete contract.

NOTE Contracts may be enforceable by law.

EXPRESS specification:

```
ENTITY contract;
  name : label;
  purpose : text;
  kind : contract_type;
END_ENTITY;
```

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.

10.3.2 contract_relationship

A **contract_relationship** is a relationship between 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.

 © ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY contract_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_contract : contract;
  related_contract : contract;
END_ENTITY;
```

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.

10.3.3 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;
```

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.

© ISO 2019 – All rights reserved

10.4 contract_schema function definitions

10.4.1 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.relating_contract IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(cont <* bag_to_set(USEDIN(relation.relating_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.relating_contract,
specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

10.4.1.1 Parameters

10.4.1.1.1 relation

(input) the candidate contract_relationship to be checked.

EXPRESS specification:

```
relation : contract_relationship
```

10.4.1.1.2 relatives

(input) the set of instances of the entity data type contract for which the function is searching in the relating_contract parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF contract
```

10.4.1.1.3 specific_relation

(input) the fully qualified entity name of a type of contract_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

 © ISO 2019 – All rights reserved

10.4.1.2 Variables

10.4.1.2.1 x

EXPRESS specification:

```
x : SET OF contract_relationship;
```

10.4.1.3 Statements

10.4.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_contract_relationship(x[i], relatives + relation.relating_contract,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

11 date_time_schema

11.1 General

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.

EXPRESS specification:

```
SCHEMA date_time_schema '{iso standard 10303 part(41) version(7) object(1) date_time_
schema(7)}';

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   description_attribute_select,
   get_description_value);

REFERENCE FROM measure_schema
  (measure_with_unit,
   time_measure_with_unit);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

11.2 Fundamental concepts and assumptions

Any kind of product data may have a date, time, or date and time associated with it.

11.3 date_time_schema type definitions

11.3.1 dts_description_attribute_select

EXPRESS specification:

```
TYPE dts_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (date_role,
   date_time_role,
   time_role);
END_TYPE;
```

© ISO 2019 – All rights reserved

11.3.2 ahead_or_behind

The **ahead_or_behind** type is used to specify whether a given time is ahead of coordinated universal time, behind coordinated universal time or is 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;
```

11.3.3 date_time_or_event_occurrence

The **date_time_or_event_occurrence** type allows for the designation of a date_time_select or an event_occurrence.

EXPRESS specification:

```
TYPE date_time_or_event_occurrence = SELECT
  (date_time_select,
   event_occurrence);
END_TYPE;
```

11.3.4 date_time_select

A **date_time_select** type allows for the designation of a date, date_and_time, or a local_time.

EXPRESS specification:

```
TYPE date_time_select = SELECT
  (date,
   date_and_time,
   local_time);
END_TYPE;
```

11.3.5 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;
```

11.3.6 day_in_week_number

A **day_in_week_number** is the position of the specified day in a week as defined in ISO 8601.

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
```

 © ISO 2019 – All rights reserved

```
  WR1: {1 <= SELF <= 7};
END_TYPE;
```

11.3.7 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;
```

11.3.8 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;
```

11.3.9 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;
```

11.3.10 month_in_year_number

A **month_in_year_number** is the position of the specified month in a year as defined in ISO 8601.

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;
```

11.3.11 second_in_minute

A **second_in_minute** is the position of the specified second in a minute.

© ISO 2019 – All rights reserved

EXPRESS specification:

```
TYPE second_in_minute = REAL;
WHERE
  WR1: {0 <= SELF < 60.0};
END_TYPE;
```

11.3.12 week_in_year_number

A **week_in_year_number** is the position of the specified calendar week as defined in ISO 8601.

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;
```

11.3.13 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;
WHERE
  WR1: SELF > 1581;
END_TYPE;
```

11.4 date_time_schema entity definitions

11.4.1 calendar_date

A **calendar_date** is a type of date defined as a day in a month of a year as defined in ISO 8601.

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;
```

Attribute definitions:

day_component the day element of the date.

month_component the month element of the date.

Attribute definitions:

WR1 The entity shall define a valid calendar date.

© ISO 2019 – All rights reserved

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.

11.4.2 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 is defined in ISO 8601 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 <= actual_minute_offset <= 59};
  WR3: NOT (((hour_offset <> 0) OR (actual_minute_offset <> 0)) AND (sense = exact));
END_ENTITY;
```

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.

Attribute definitions:

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.

11.4.3 date

A **date** is the identification of a day or week or month in a year. A **date** may be a calendar_date or an ordinal_date, or a week_of_year_and_day_date, or a year_month. The date is defined in ISO 8601.

© ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY date
  SUPERTYPE OF (ONEOF(calendar_date, ordinal_date, week_of_year_and_day_date, year_
month));
  year_component : year_number;
END_ENTITY;
```

Attribute definitions:

year_component the year in which the **date** occurs.

11.4.4 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;
```

Attribute definitions:

date_component the date element of the date time combination.

time_component the time element of the date time combination.

11.4.5 date_role

A **date_role** is a role for a date A **date_role** is an incompletely modeled concept and requires population related entity data types to form a complete **date_role**.

NOTE 1 A **date_role** is 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;
```

Attribute definitions:

name the label by which the **date_role** 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 **date_role**. The value of the attribute need not be specified.

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

© ISO 2019 – All rights reserved

Attribute definitions:

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

NOTE 4 The description_attribute data type is defined in basic_attribute_schema 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.

11.4.6 date_time_role

A **date_time_role** is a role for a moment of time on a particular day. A **date_time_role** is an incompletely modeled concept that requires population of related entity data types to form a complete **date_time_role**

NOTE 1 A **date_time_role** is 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;
```

Attribute definitions:

name the label by which the **date_time_role** 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 **date_time_role**.

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

Attribute definitions:

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

NOTE 4 The description_attribute data type is defined in basic_attribute_schema 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.

11.4.7 event_occurrence

An **event_occurrence** is the fact of a transition between states at some point in time. The description of the **event_occurrrnece** can provide an implicit representation of either the state preceeding the **event_occurrence** or the state after the **event_occurrence**.

© ISO 2019 – All rights reserved

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 The **event_occurrence** 'start of production' implicitly declares that the before the **event_occurrence** a product was not in production.

EXPRESS specification:

```
ENTITY event_occurrence;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

11.4.8 event_occurrence_context_role

An **event_occurrence_context_role** establishes a context for an event_occurrence_assignment. An **event_occurrence** is an incompletely modeled concept and requires population of related entity data types to form a complete **event_occurrence_context_role**

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;
```

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.

11.4.9 event_occurrence_relationship

An **event_occurrence_relationship** is a relationship between two event_occurrences with a description of their relationship.

 © ISO 2019 – All rights reserved

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;
```

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.

11.4.10 event_occurrence_role

An **event_occurrence_role** establishes a role for an **event_occurrence**. An **event_occurrence_role** is an incompletely modeled concept and requires population of related entity data types to form a complete **event_occurrence_role**

EXPRESS specification:

```
ENTITY event_occurrence_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

© ISO 2019 – All rights reserved

11.4.11 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. This concept is defined in ISO 8601.

NOTE This construct is used to represent a moment of time whereas time_measure represents amounts 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;
```

Attribute definitions:

hour_component the number of hours.

minute_component the number of minutes. The value of the attribute need not be specified.

second_component the number of seconds. The value of the attribute need not be specified.

zone the offset of the local time zone to the coordinated universal time.

Attribute definitions:

WR1 The entity shall define a valid time. The **second_component** attribute shall only exist if the **minute_component** attribute exists.

11.4.12 ordinal_date

An **ordinal_date** is a type of date defined as a day of a year. This concept is defined in ISO 8601.

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;
```

Attribute definitions:

day_component the day element of the date.

Attribute definitions:

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.

© ISO 2019 – All rights reserved

11.4.13 relative_event_occurrence

A **relative_event_occurrence** is a type of event_occurrence that takes place with an offset to an 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;
```

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**.

11.4.14 time_interval

A **time_interval** is the identification of an interval of time. This concept is defined in ISO 8601.

EXAMPLE 'Strike duration', 'delay of production', and 'Christmas holidays' are examples for *time_interval*s.

EXPRESS specification:

```
ENTITY time_interval;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

11.4.15 time_interval_relationship

A **time_interval_relationship** is a relationship between two time_intervals with a description of their relationship.

© ISO 2019 – All rights reserved

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;
```

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.

11.4.16 time_interval_role

A **time_interval_role** is a role that may be assigned to a **time_interval** and includes a description of that role.

NOTE 1 The assignment of a role to a **time_interval** is accomplished by the **time_interval_assignment** entity data type.

EXPRESS specification:

```
ENTITY time_interval_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

© ISO 2019 – All rights reserved

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.

11.4.17 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_interval_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: EXISTS(primary_bound) OR EXISTS(secondary_bound);
  WR2: NOT ((EXISTS(primary_bound) AND EXISTS(secondary_bound)) AND EXISTS(duration));
  WR3: (EXISTS(primary_bound) AND NOT EXISTS(secondary_bound)) AND (NOT EXISTS(duration)
OR duration_is_positive(duration));
  WR4: ((EXISTS(secondary_bound) AND NOT EXISTS(primary_bound)) AND EXISTS(duration)) AND
duration_is_negative(duration);
END_ENTITY;
```

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.

Attribute definitions:

WR1 Either the **primary_bound** or the **secondary_bound** or both shall be specified.

WR2 The combination of **primary_bound**, **secondary_bound** and **duration** shall not be specified.

WR3 If **primary_bound** is specified and **secondary_bound** is not specified and if **duration** is specified, the **duration** shall be positive.

WR4 If **primary_bound** is not specified and **secondary_bound** is specified, **duration** shall be specified and the **duration** shall be negative.

11.4.18 time_role

A **time_role** is a definition of a role for a **local_time**.

NOTE 1 The assignment of the role to a **local_time** is accomplished by the **time_assignment** entity data type.

© ISO 2019 – All rights reserved

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;
```

Attribute definitions:

name the label by which the **time_role** 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 **time_role**.

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

Attribute definitions:

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

NOTE 4 The description_attribute data type is defined in basic_attribute_schema 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.

11.4.19 week_of_year_and_day_date

A **week_of_year_and_day_date** is a type of 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;
WHERE
  WR1: NOT leap_year(SELF\date.year_component) OR {1 <= day_component + 7 * (week_
component - 1) <= 366};
  WR2: leap_year(SELF\date.year_component) OR {1 <= day_component + 7 * (week_component -
1) <= 365};
END_ENTITY;
```

Attribute definitions:

week_component the week element of the date.

day_component the day element of the date.

Attribute definitions:

© ISO 2019 – All rights reserved

WR1 The result of the expression **day_component** + (7 * week_component) shall be between 1 and 366 if the year_component is a leap year.

WR2 The result of the expression **day_component** + (7 * week_component) shall be between 1 and 365 if the year_component is not a leap year.

11.4.20 year_month

A **year_month** is a type of date defined as a month of a year.

EXPRESS specification:

```
ENTITY year_month
  SUBTYPE OF (date);
  month_component : month_in_year_number;
END_ENTITY;
```

Attribute definitions:

month_component the month element of the date.

11.5 date_time_schema function definitions

11.5.1 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 an 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;
```

© ISO 2019 – All rights reserved

11.5.1.1 Parameters

11.5.1.1.1 relation

(input) the candidate event_occurrence_relationship to be checked.

EXPRESS specification:

```
relation : event_occurrence_relationship
```

11.5.1.1.2 relatives

(input) the set of instances of the entity data type event_occurrence for which the function is searching in the relating_event parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF event_occurrence
```

11.5.1.1.3 specific_relation

(input) the fully qualified entity name of a type of event_occurrence_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

11.5.1.2 Variables

11.5.1.2.1 x

EXPRESS specification:

```
x : SET OF event_occurrence_relationship;
```

11.5.1.3 Statements

11.5.1.3.1 i

EXPRESS specification:

```
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;
```

11.5.2 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;
```

 © ISO 2019 – All rights reserved

```
                                                specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF time_interval_relationship;
  END_LOCAL;
  IF relation.relating_time_interval IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(ti <* bag_to_set(USEDIN(relation.relating_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.relating_time_
interval, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

11.5.2.1 Parameters

11.5.2.1.1 relation

(input) the candidate time_interval_relationship to be checked.

EXPRESS specification:

```
relation : time_interval_relationship
```

11.5.2.1.2 relatives

(input) the set of instances of the entity data type time_interval for which the function is searching in the relating_time_interval parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF time_interval
```

11.5.2.1.3 specific_relation

(input) the fully qualified entity name of a type of time_interval_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

11.5.2.2 Variables

11.5.2.2.1 x

EXPRESS specification:

```
x : SET OF time_interval_relationship;
```

11.5.2.3 Statements

11.5.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_time_interval_relationship(x[i], relatives + relation.relating_time_
interval, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
```

© ISO 2019 – All rights reserved

```
END_REPEAT;
```

11.5.3 duration_is_positive

The **duration_is_positive** function determines whether a given duration is positive. It returns TRUE if the duration is a positive. Otherwise, it returns FALSE.

EXPRESS specification:

```
FUNCTION duration_is_positive(duration : time_measure_with_unit) : BOOLEAN;
  IF duration\measure_with_unit.value_component > 0.0 THEN
    RETURN (TRUE);
  ELSE
    RETURN (FALSE);
  END_IF;
END_FUNCTION;
```

11.5.3.1 Parameters

11.5.3.1.1 duration

(input) the candidate duration that is being checked.

EXPRESS specification:

```
duration : time_measure_with_unit
```

11.5.4 duration_is_negative

The **duration_is_negative** function determines whether a given duration is negative. It returns TRUE if the duration is negative. Otherwise, it returns FALSE.

EXPRESS specification:

```
FUNCTION duration_is_negative(duration : time_measure_with_unit) : BOOLEAN;
  IF duration\measure_with_unit.value_component < 0.0 THEN
    RETURN (TRUE);
  ELSE
    RETURN (FALSE);
  END_IF;
END_FUNCTION;
```

11.5.4.1 Parameters

11.5.4.1.1 duration

(input) the candidate duration that is being checked.

EXPRESS specification:

```
duration : time_measure_with_unit
```

11.5.5 leap_year

The **leap_year** function checks whether or not a particular year contains 366 days instead of 365 days, known as a leap year.

NOTE Leap years occur every for years.

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
```

 © ISO 2019 – All rights reserved

```
    RETURN (FALSE);
  END_IF;
END_FUNCTION;
```

11.5.5.1 Parameters

11.5.5.1.1 year

an integer representing a particular year in the Gregorian calendar.

EXPRESS specification:

```
year : year_number
```

11.5.6 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;
```

11.5.6.1 Parameters

11.5.6.1.1 date

(input) the candidate calendar_date that is to be checked.

EXPRESS specification:

© ISO 2019 – All rights reserved

```
date : calendar_date
```

11.5.7 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;
```

11.5.7.1 Parameters

11.5.7.1.1 time

(input) the candidate local_time that is to be checked.

EXPRESS specification:

```
time : local_time
```

12 document_schema

12.1 General

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.

EXPRESS specification:

```
SCHEMA document_schema '{iso standard 10303 part(41) version(4) object(1) document_
schema(8)}';

REFERENCE FROM product_definition_schema
  (product,
   product_definition,
   product_definition_formation);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

12.2 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.

 © ISO 2019 – All rights reserved

12.3 document_schema type definitions

12.3.1 product_or_formation_or_definition

The **product_or_formation_or_definition** type allows for the designation of a product, a product_definition, or a product_definition_formation.

EXPRESS specification:

```
TYPE product_or_formation_or_definition = SELECT
  (product,
   product_definition_formation,
   product_definition);
END_TYPE;
```

12.4 document_schema entity definitions

12.4.1 document

A **document** is an identification of a collection of information.

NOTE 1 The information collected need not be represented in a format conforming to any EXPRESS schema.

EXAMPLE 1 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;
```

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**.

EXAMPLE 2 An example of the source of a **document** is “ISO”.

© ISO 2019 – All rights reserved

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.

12.4.2 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;
```

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.

12.4.3 document_relationship

A **document_relationship** is a relationship between 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.

 © ISO 2019 – All rights reserved

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;
```

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.

12.4.4 document_representation_type

A **document_representation_type** is a specification of the way a document is represented.

EXPRESS specification:

```
ENTITY document_representation_type;
  name : label;
  represented_document : document;
END_ENTITY;
```

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.

12.4.5 document_type

A **document_type** is the kind of data that a formal standard or document may provide information about.

© ISO 2019 – All rights reserved

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;
```

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.

12.4.6 document_usage_constraint

A **document_usage_constraint** is an identification of 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;
```

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'.

12.4.7 document_with_class

A **document_with_class** is a type of document for which a classification is specified.

 © ISO 2019 – All rights reserved

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;
```

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.

12.5 document_schema function definitions

12.5.1 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;
```

© ISO 2019 – All rights reserved

12.5.1.1 Parameters

12.5.1.1.1 relation

(input) the candidate document_relationship to be checked.

EXPRESS specification:

```
relation : document_relationship
```

12.5.1.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF document
```

12.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the document_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

12.5.1.2 Variables

12.5.1.2.1 x

EXPRESS specification:

```
x : SET OF document_relationship;
```

12.5.1.3 Statements

12.5.1.3.1 i

EXPRESS specification:

```
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;
```

13 effectivity_schema

13.1 General

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.

EXPRESS specification:

 © ISO 2019 – All rights reserved

```
SCHEMA effectivity_schema '{iso standard 10303 part(41) version(8) object(1) effectivity_
schema(9)}';

REFERENCE FROM action_schema
  (action_method);

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   description_attribute_select,
   get_description_value,
   get_name_value,
   name_attribute,
   name_attribute_select);

REFERENCE FROM date_time_schema
  (date_time_or_event_occurrence,
   time_interval);

REFERENCE FROM measure_schema
  (measure_with_unit);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

13.2 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.

13.3 effectivity_schema type definitions

13.3.1 effs_description_attribute_select

EXPRESS specification:

```
TYPE effs_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (effectivity);
END_TYPE;
```

13.3.2 effs_name_attribute_select

EXPRESS specification:

```
TYPE effs_name_attribute_select = SELECT BASED_ON name_attribute_select WITH
  (effectivity);
END_TYPE;
```

13.4 effectivity_schema entity definitions

13.4.1 condition

A **condition** is a type of action_method that must be fulfilled before a statement or relationship becomes valid.

EXPRESS specification:

```
ENTITY condition
  SUBTYPE OF (action_method);
END_ENTITY;
```

© ISO 2019 – All rights reserved

13.4.2 conditional_effectivity

A **conditional_effectivity** is a type of condition or effectivity for which the domain of applicability is defined by an explicit condition.

EXPRESS specification:

```
ENTITY conditional_effectivity
  SUBTYPE OF (condition, effectivity);
END_ENTITY;
```

13.4.3 conditional_configuration

A **conditional_configuration** is a type of conditional_effectivity and represents an association of a product_concept_feature_association object with a design or process in order to define a valid usage of it in the context of a certain product_class.

EXPRESS specification:

```
ENTITY conditional_configuration
  SUBTYPE OF (conditional_effectivity);
  configuration_type : STRING;
  inheritance_type : STRING;
END_ENTITY;
```

Attribute definitions:

configuration_type the **configuration_type** defines the valid usage of a **conditional_configuration**.

inheritance_type the **inheritance_type** specifies the inheritance scheme for the configuration information as either **exception**, **inherited**, or **local**.

- An **exception** inheritance_type is when no inheritance scheme is applicable and all required configuration information must be attached locally at the business object.
- An **inherited** inheritance_type is when a scheme for inheritance of configuration information applies. The complete configuration information shall be collected from the different levels in the structure by evaluation of results.
- A **local** inheritance_type is when no inheritance scheme is applicable and all required configuration information must be attached locally at the business object. Nevertheless, any potentially inherited configuration information of a higher level shall be consistent, i.e., it shall be a subset of the locally defined configuration information.

13.4.4 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 : OPTIONAL date_time_or_event_occurrence;
WHERE
  WR1: EXISTS(effectivity_end_date) OR EXISTS(effectivity_start_date);
```


© ISO 2019 – All rights reserved

```
END_ENTITY;
```

Attribute definitions:

effectivity_end_date the date_time_or_event_occurrence 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.

effectivity_start_date the date_time_or_event_occurrence that defines the lower bound of the interval of applicability. If a value for this attribute is not defined, the interval of applicability has no lower limit.

Attribute definitions:

WR1 One or both of the **effectivity_start_date** or **effectivity_end_date** shall be specified.

13.4.5 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.

An **effectivity** may be either a serial_numbered_effectivity or a dated_effectivity, or a lot_effectivity, or a time_interval_based_effectivity.

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;
```

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**.

© ISO 2019 – All rights reserved

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

Attribute definitions:

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

NOTE 5 The description_attribute data type is defined in basic_attribute_schema 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 basic_attribute_schema 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.

13.4.6 effectivity_relationship

An **effectivity_relationship** is a relationship between 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;
```

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.

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 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.

relating_effectivity one of the instances of effectivity that is a part of the relationship.

 © ISO 2019 – All rights reserved

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.

13.4.7 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;
```

Attribute definitions:

effectivity_lot_id the identification of the batch of items.

effectivity_lot_size the size of the batch of items.

13.4.8 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 of 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;
```

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.

13.4.9 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;
```

© ISO 2019 – All rights reserved

```
END_ENTITY;
```

Attribute definitions:

effectivity_period the time_interval defining the period of time when the associated product data are effective.

13.5 effectivity_schema function definitions

13.5.1 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.relating_effectivity IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(eff <* bag_to_set(USEDIN(relation.relating_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.relating_
effectivity, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

13.5.1.1 Parameters

13.5.1.1.1 relation

(input) the candidate effectivity_relationship to be checked.

EXPRESS specification:

```
relation : effectivity_relationship
```

13.5.1.1.2 relatives

(input) the set of instances of the entity data type effectivity for which the function is searching in the relating_effectivity parameter of the **relation** argument.

EXPRESS specification:

 © ISO 2019 – All rights reserved

```
relatives : SET [1:?] OF effectivity
```

13.5.1.1.3 specific_relation

(input) the fully qualified entity name of a type of effectivity_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

13.5.1.2 Variables

13.5.1.2.1 x

EXPRESS specification:

```
x : SET OF effectivity_relationship;
```

13.5.1.3 Statements

13.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_effectivity_relationship(x[i], relatives + relation.relating_
effectivity, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

14 experience_schema

14.1 General

The subject of the **experience_schema** is the identification and characterization of types of experience or particular episodes of experience gained by persons. This schema provides for the definition of generic data relating to particular episodes of experience.

EXAMPLE 1 The identification of a particular episode of experience is an example of generic data relating to experience.

EXAMPLE 2 The identification of a particular type of experience is an example of generic data relating to experience.

EXPRESS specification:

```
SCHEMA experience_schema '{iso standard 10303 part(41) version(4) object(1) experience_
schema(10)}';

REFERENCE FROM support_resource_schema
  (identifier,
   label,
   text,
   bag_to_set);
```

14.2 Fundamental concepts and assumptions

This schema provides resources for the identification, naming, and description of:

— experience;

— types of experience;

© ISO 2019 – All rights reserved

— relationships between specific experiences;

— relationships between types of experience.

14.3 experience_schema entity definitions

14.3.1 experience

An **experience** is an episode of practice, undertaken by a person or organization, that contributes to the accumulation of knowledge or a skill.

EXAMPLE 100 flying hours in a Tornado jet undertaken by a particular pilot in a particular year.

EXPRESS specification:

```
ENTITY experience;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience**.

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 **experience** is known.

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.

description the text that characterizes the **experience**. The value of this attribute need not be specified.

14.3.2 experience_relationship

An **experience_relationship** is a relationship between two instances of the experience entity data type and provides for the identification, naming, and description of that relationship.

EXPRESS specification:

```
ENTITY experience_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_experience : experience;
  related_experience : experience;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience_relationship**.

© ISO 2019 – All rights reserved

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 **experience_relationship** is known.

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.

description the text that characterizes the **experience_relationship**. The value of this attribute need not be specified.

relating_experience one of the instances of experience 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_experience the other instance of experience 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.

14.3.3 experience_type

An **experience_type** is a category of experience.

EXAMPLE Flying experience.

NOTE 1 A relationship between an **experience_type** object and one or more experience objects is established by the declaration of an experience_type_assignment subtype containing a single attribute that references a set of a SELECT type that contains an experience.

EXPRESS specification:

```
ENTITY experience_type;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience_type**.

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.

© ISO 2019 – All rights reserved

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.

name the label by which the **experience_type** is known.

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.

description the text that characterizes the **experience_type**. The value of this attribute need not be specified.

14.3.4 experience_type_relationship

An **experience_type_relationship** is a relationship between two instances of the experience_type entity data type and provides for the identification, naming, and description of that relationship.

EXPRESS specification:

```
ENTITY experience_type_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_experience_type : experience_type;
  related_experience_type : experience_type;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience_type_relationship**.

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 **experience_type_relationship** is known.

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.

description the text that characterizes the **experience_type_relationship**. The value of this attribute need not be specified.

relating_experience_type one of the instances of experience_type 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.

© ISO 2019 – All rights reserved

related_experience_type the other instance of experience_type 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.

14.4 experience_schema function definitions

14.4.1 acyclic_experience_relationship

The **acyclic_experience_relationship** function determines whether the graph of instances of the entity data type experience that contains relation as one of its links contains a cycle. This function may be used to evaluate either an experience_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 experience_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_experience_relationship(relation : experience_relationship;
                                         relatives : SET OF experience;
                                         specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF experience_relationship;
  END_LOCAL;
  IF relation.relating_experience IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(expr <* bag_to_set(USEDIN(relation.relating_experience, ('EXPERIENCE_
SCHEMA.' + 'EXPERIENCE_RELATIONSHIP.') + 'RELATED_EXPERIENCE')) | specific_relation IN
TYPEOF(expr));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_experience_relationship(x[i], relatives + relation.relating_
experience, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

14.4.1.1 Parameters

14.4.1.1.1 relation

(input) the candidate experience_relationship to be checked.

EXPRESS specification:

```
relation : experience_relationship
```

14.4.1.1.2 relatives

(input) the set of instances of the entity data type experience that the function is searching for in the relating_experience parameter of the **relation** argument.

© ISO 2019 – All rights reserved

EXPRESS specification:

```
relatives : SET OF experience
```

14.4.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the experience_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

14.4.1.2 Variables

14.4.1.2.1 x

EXPRESS specification:

```
x : SET OF experience_relationship;
```

14.4.1.3 Statements

14.4.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_experience_relationship(x[i], relatives + relation.relating_experience,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

14.4.2 acyclic_experience_type_relationship

The **acyclic_experience_type_relationship** function determines whether the graph of instances of the entity data type experience_type that contains relation as one of its links contains a cycle. This function may be used to evaluate either an experience_type_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 experience_type_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_experience_type_relationship(relation : experience_type_relationship;
                                              relatives : SET OF experience_type;
                                              specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF experience_type_relationship;
  END_LOCAL;
  IF relation.relating_experience_type IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(exptyp <* bag_to_set(USEDIN(relation.relating_experience_type, ('EXPERIENCE_
SCHEMA.' + 'EXPERIENCE_TYPE_RELATIONSHIP.') + 'RELATED_EXPERIENCE_TYPE')) | specific_
relation IN TYPEOF(exptyp));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_experience_type_relationship(x[i], relatives + relation.relating_
experience_type, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
```

 © ISO 2019 – All rights reserved

```
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

14.4.2.1 Parameters

14.4.2.1.1 relation

(input) the candidate experience_type_relationship to be checked.

EXPRESS specification:

```
relation : experience_type_relationship
```

14.4.2.1.2 relatives

(input) the set of instances of the entity data type experience_type that the function is searching for in the relating_experience_type parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET OF experience_type
```

14.4.2.1.3 specific_relation

(input) the fully qualified name of a subtype of the experience_type_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

14.4.2.2 Variables

14.4.2.2.1 x

EXPRESS specification:

```
x : SET OF experience_type_relationship;
```

14.4.2.3 Statements

14.4.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_experience_type_relationship(x[i], relatives + relation.relating_
experience_type, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

15 external_reference_schema

15.1 General

The subject of the **external_reference_schema** is the means of identifying information that is not explicitly represented in a given exchange.

EXPRESS specification:

```
SCHEMA external_reference_schema '{iso standard 10303 part(41) version(7) object(1)
external_reference_schema(11)}';

REFERENCE FROM basic_attribute_schema
```

© ISO 2019 – All rights reserved

```
  (description_attribute,
   description_attribute_select,
   get_description_value);

REFERENCE FROM support_resource_schema
  (label,
   text,
   identifier,
   bag_to_set);
```

15.2 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 entity data type definitions in the annotated EXPRESS schema. Otherwise, this requirement is satisfied by a reference that identifies the record type definitions and exchange format.

15.3 external_reference_schema type definitions

15.3.1 ers_description_attribute_select

EXPRESS specification:

```
TYPE ers_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (external_source);
END_TYPE;
```

15.3.2 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;
```

15.3.3 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;
```

15.4 external_reference_schema entity definitions

15.4.1 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.

 © ISO 2019 – All rights reserved

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;
```

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.

Attribute definitions:

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 basic_attribute_schema 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.

15.4.2 external_source_relationship

An **external_source_relationship** is a relationship between 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;
```

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.

© ISO 2019 – All rights reserved

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.

15.4.3 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;
```

Attribute definitions:

item_id the string that specifies the identification of the referent item.

source an external_source that contains the referent item.

15.4.4 externally_defined_item_relationship

An **externally_defined_item_relationship** is a relationship between 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;
```

Attribute definitions:

name the label by which the **externally_defined_item_relationship** is known.

© ISO 2019 – All rights reserved

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

relating_item one of the instances 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.

15.4.5 externally_defined_item_with_multiple_references

An **externally_defined_item_with_multiple_references** is a type of externally_defined_item that allows to define additional references providing information on the context and location of the target item within the source.

EXPRESS specification:

```
ENTITY externally_defined_item_with_multiple_references
  SUBTYPE OF (externally_defined_item);
  references : LIST [1:?] OF UNIQUE source_item;
UNIQUE
  UR1: references;
WHERE
  WR1: NOT (SELF\externally_defined_item.item_id IN references);
END_ENTITY;
```

Attribute definitions:

references a list of source_items defining a path to the target item_id.

Attribute definitions:

WR1 An **externally_defined_item_with_multiple_references** is a type of item_id that adds a list of references to provide an unambiguous path in the external data. The item_id inherited from the externally_defined_item is the target for the reference list, but is excluded from the list itself. The list ends at a entity data type that explicitly references the target or that is explicitly referenced by the target.

15.4.6 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;
```

© ISO 2019 – All rights reserved

```
END_ENTITY;
```

Attribute definitions:

name the label by which the **pre_defined_item** is known.

15.5 external_reference_schema function definitions

15.5.1 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 an 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.relating_source IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(es <* bag_to_set(USEDIN(relation.relating_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.relating_
source, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

15.5.1.1 Parameters

15.5.1.1.1 relation

(input) the candidate external_source_relationship to be checked.

EXPRESS specification:

```
relation : external_source_relationship
```

15.5.1.1.2 relatives

(input) the set of instances of the entity data type external_source for which the function is searching in the relating_source parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET [1:?] OF external_source
```


© ISO 2019 – All rights reserved

15.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the external_source_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

15.5.1.2 Variables

15.5.1.2.1 x

EXPRESS specification:

```
x : SET OF external_source_relationship;
```

15.5.1.3 Statements

15.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_external_source_relationship(x[i], relatives + relation.relating_source,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

15.5.2 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 an 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.relating_item IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(edi <* bag_to_set(USEDIN(relation.relating_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.
relating_item, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
```

© ISO 2019 – All rights reserved

```
  RETURN (TRUE);
END_FUNCTION;
```

15.5.2.1 Parameters

15.5.2.1.1 relation

(input) the candidate externally_defined_item_relationship to be checked.

EXPRESS specification:

```
relation : externally_defined_item_relationship
```

15.5.2.1.2 relatives

(input) the set of instances of the entity data type externally_defined_item for which the function is searching in the relating_item parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET [1:?] OF externally_defined_item
```

15.5.2.1.3 specific_relation

(input) the fully qualified entity name of a type of externally_defined_item_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

15.5.2.2 Variables

15.5.2.2.1 x

EXPRESS specification:

```
x : SET OF externally_defined_item_relationship;
```

15.5.2.3 Statements

15.5.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_externally_defined_item_relationship(x[i], relatives + relation.relating_
item, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

16 group_schema

16.1 General

The subject of the **group_schema** is the definition of groups of items.

EXPRESS specification:

```
SCHEMA group_schema '{iso standard 10303 part(41) version(7) object(1) group_schema(12)}';

REFERENCE FROM support_resource_schema
  (label,
   bag_to_set,
   identifier,
```

 © ISO 2019 – All rights reserved

```
   text);

REFERENCE FROM basic_attribute_schema
  (get_id_value,
   id_attribute,
   id_attribute_select);
```

16.2 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.

16.3 group_schema type definitions

16.3.1 gs_id_attribute_select

EXPRESS specification:

```
TYPE gs_id_attribute_select = SELECT BASED_ON id_attribute_select WITH
  (group);
END_TYPE;
```

16.4 group_schema entity definitions

16.4.1 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;
```

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.

© ISO 2019 – All rights reserved

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.

Attribute definitions:

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

NOTE 5 The id_attribute data type is defined in basic_attribute_schema 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.

16.4.2 group_relationship

A **group_relationship** is a relationship between 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;
```

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.

© ISO 2019 – All rights reserved

16.5 group_schema function definitions

16.5.1 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;
```

16.5.1.1 Parameters

16.5.1.1.1 relation

(input) the candidate group_relationship to be checked.

EXPRESS specification:

```
relation : group_relationship
```

16.5.1.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF group
```

16.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the group_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

© ISO 2019 – All rights reserved

16.5.1.2 Variables

16.5.1.2.1 x

EXPRESS specification:

```
x : SET OF group_relationship;
```

16.5.1.3 Statements

16.5.1.3.1 i

EXPRESS specification:

```
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;
```

17 language_schema

17.1 General

The subject of the **language_schema** specifies the data structure for the representation of language support.- IP1: The name attribute of the language shall be an alpha-3 bibliographic code of ISO 639-2.- IP2: If present, the description attribute shall identify a country with an alpha-2 code specified in ISO 3166-1.

EXPRESS specification:

```
SCHEMA language_schema '{iso standard 10303 part(41) version(1) object(1) location_
schema(13)}';

REFERENCE FROM group_schema
  (group);
```

17.2 Fundamental concepts and assumptions

This specifies the representation of the language used in text attributes and the association of translations to these texts.

17.3 language_schema entity definitions

17.3.1 language

A **language** is a type of group that is a specification of the language in which an information is given. The group shall specify in its name a language code according to ISO 639-2, and, if present, in its description a country code according to ISO 3166-1.Informal proposition

EXPRESS specification:

```
ENTITY language
  SUBTYPE OF (group);
WHERE
  WR1: SELF\group.name <> '';
END_ENTITY;
```

Attribute definitions:

WR1 The name attribute of the language shall be specified.

© ISO 2019 – All rights reserved

18 location_schema

18.1 General

The subject of the **location_schema** is the identification and characterization of different representations of places or positions in space at which something can take place or exist.

EXAMPLE The identification of a geographical location is an example of generic data relating to a location.

EXPRESS specification:

```
SCHEMA location_schema '{iso standard 10303 part(41) version(4) object(1) location_
schema(14)}';

REFERENCE FROM support_resource_schema
  (identifier,
   label,
   text,
   bag_to_set);
```

18.2 Fundamental concepts and assumptions

This schema provides resources for the identification, naming, and description of:

— specific locations;

— relationships between specific locations.

18.3 location_schema entity definitions

18.3.1 location

A **location** is a place or position in space at which something can take place or exist.

NOTE The specification or representation of a location might take the form of a postal address, a set of geographical or spacial coordinates, or an association with a product.

EXAMPLE 1 A map reference specifies a location where a building exists.

EXAMPLE 2 The “bridge” of a particular ship specifies a location where navigation equipment should be installed.

EXPRESS specification:

```
ENTITY location;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **location**.

name the label by which the **location** is known.

description the text that characterizes the **location**. The value of this attribute need not be specified.

© ISO 2019 – All rights reserved

18.3.2 location_relationship

A **location_relationship** is a relationship between two instances of the location entity data type and provides for the identification, naming, and description of that relationship.

EXPRESS specification:

```
ENTITY location_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_location : location;
  related_location : location;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **location_relationship**.

name the label by which the **location_relationship** is known.

description the text that characterizes the **location_relationship**. The value of this attribute need not be specified.

relating_location one of the instances of location that is a part of the relationship.

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.

related_location the other instance of location 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 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.

18.4 location_schema function definitions

18.4.1 acyclic_location_relationship

The **acyclic_location_relationship** function determines whether the graph of instances of the entity data type location that contains relation as one of its links contains a cycle. This function may be used to evaluate either a location_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 location_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_location_relationship(relation : location_relationship;
                                       relatives : SET OF location;
                                       specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF location_relationship;
  END_LOCAL;
  IF relation.relating_location IN relatives THEN
```

 © ISO 2019 – All rights reserved

```
    RETURN (FALSE);
  END_IF;
  x := QUERY(lctn <* bag_to_set(USEDIN(relation.relating_location, ('LOCATION_SCHEMA.' +
'LOCATION_RELATIONSHIP.') + 'RELATED_LOCATION')) | specific_relation IN TYPEOF(lctn));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_location_relationship(x[i], relatives + relation.relating_location,
specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

18.4.1.1 Parameters

18.4.1.1.1 relation

(input) the candidate location_relationship to be checked.

EXPRESS specification:

```
relation : location_relationship
```

18.4.1.1.2 relatives

(input) the set of instances of the entity data type location that the function is searching for in the relating_location parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET OF location
```

18.4.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the location_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

18.4.1.2 Variables

18.4.1.2.1 x

EXPRESS specification:

```
x : SET OF location_relationship;
```

18.4.1.3 Statements

18.4.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_location_relationship(x[i], relatives + relation.relating_location,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

© ISO 2019 – All rights reserved

19 management_resources_schema

19.1 General

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

EXPRESS specification:

```
SCHEMA management_resources_schema '{iso standard 10303 part(41) version(8) object(1)
management_resources_schema(15)}';

REFERENCE FROM action_schema;

REFERENCE FROM application_context_schema
  (library_context);

REFERENCE FROM approval_schema;

REFERENCE FROM basic_attribute_schema
  (get_role,
   object_role,
   role_association,
   role_select);

REFERENCE FROM certification_schema;

REFERENCE FROM contract_schema;

REFERENCE FROM date_time_schema;

REFERENCE FROM document_schema;

REFERENCE FROM effectivity_schema;

REFERENCE FROM experience_schema;

REFERENCE FROM external_reference_schema
  (external_source);

REFERENCE FROM group_schema;

REFERENCE FROM location_schema;

REFERENCE FROM person_organization_schema;

REFERENCE FROM qualifications_schema;

REFERENCE FROM security_classification_schema;

REFERENCE FROM support_resource_schema;
```

19.2 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.5.

 © ISO 2019 – All rights reserved

19.3 management_resources_schema type definitions

19.3.1 mrs_role_select

EXPRESS specification:

```
TYPE mrs_role_select = SELECT BASED_ON role_select WITH
  (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);
END_TYPE;
```

19.3.2 assignment_object_select

EXPRESS specification:

```
TYPE assignment_object_select = EXTENSIBLE SELECT;
END_TYPE;
```

19.3.3 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;
```

19.4 management_resources_schema entity definitions

19.4.1 action_assignment

An **action_assignment** is an association of an action 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;
```

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.

© ISO 2019 – All rights reserved

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.

Attribute definitions:

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.

19.4.2 action_method_assignment

An **action_method_assignment** is an association of an action_method with product data.

EXPRESS specification:

```
ENTITY action_method_assignment
  ABSTRACT SUPERTYPE;
  assigned_action_method : action_method;
  role : action_method_role;
END_ENTITY;
```

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.

19.4.3 action_method_role

An **action_method_role** is a definition of a role for an action_method 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;
```

Attribute definitions:

name the label by which the **action_method_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.

 © ISO 2019 – All rights reserved

NOTE 2 The entity data type action_method_assignment implements the association of the role with a specific action_method.

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

19.4.4 action_request_assignment

An **action_request_assignment** is an association of a versioned_action_request 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;
```

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.

Attribute definitions:

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.

19.4.5 approval_assignment

An **approval_assignment** is an association of an approval 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;
```

© ISO 2019 – All rights reserved

```
END_ENTITY;
```

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.

Attribute definitions:

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.

19.4.6 assignment_object_relationship

An **assignment_object_relationship** is a relationship between two objects of the assignment_object_select list representing the same object in different organizational contexts.

EXPRESS specification:

```
ENTITY assignment_object_relationship;
  id : OPTIONAL identifier;
  description : OPTIONAL text;
  relating : assignment_object_select;
  related : assignment_object_select;
  relation_type : STRING;
WHERE
  WR1: acyclic_assignment_object_relationship(SELF, [related], 'MANAGEMENT_RESOURCES_
SCHEMA.ASSIGNMENT_OBJECT_RELATIONSHIP');
END_ENTITY;
```

Attribute definitions:

id the identifier or set of identifiers for the **assignment_object_relationship**.

description the text or the set of texts that provide further information about the **assignment_object_relationship**. The value of this attribute need not be specified.

relating one object of assignment_object_select that is part of the relationship.

related the other object of assignment_object_select that is part of the relationship.

relation_type the meaning of the relationship.

Attribute definitions:

WR1 An **assignment_object_relationship** shall not participate in its own definition.

 © ISO 2019 – All rights reserved

19.4.7 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 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.

19.4.8 attribute_value_assignment

An **attribute_value_assignment** is an extendion of the cardinality of a string valued attribute in an entity data type by one. The relationship of the extension to the existing attribute is determined by the role. The entity data types that have the attribute for which the extension is provided are identified by a specialization 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 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.

© ISO 2019 – All rights reserved

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.

19.4.9 attribute_value_role

An **attribute_value_role** is a description of 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 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.

19.4.10 certification_assignment

A **certification_assignment** is an association of a certification 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;
```

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.

Attribute definitions:

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.

 © ISO 2019 – All rights reserved

19.4.11 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;
```

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.

19.4.12 classification_assignment_relationship

A **classification_assignment_relationship** is a relationship between two classification_assignment objects.

EXPRESS specification:

```
ENTITY classification_assignment_relationship;
  id : OPTIONAL identifier;
  description : OPTIONAL text;
  relating : classification_assignment;
  related : classification_assignment;
  relation_type : STRING;
WHERE
  WR1: acyclic_classification_assignment_relationship(SELF, [related], 'MANAGEMENT_
RESOURCES_SCHEMA.CLASSIFICATION_ASSIGNMENT_RELATIONSHIP');
END_ENTITY;
```

Attribute definitions:

id the identifier or set of identifiers for the **classification_assignment_relationship**.

description the text or the set of texts that provide further information about the **classification_assignment_relationship**. The value of this attribute need not be specified.

relating one object of classification_assignment that is part of the relationship.

related the other object of classification_assignment that is part of the relationship.

relation_type the meaning of the relationship.

Attribute definitions:

WR1 A **classification_assignment_relationship** shall not participate in its own definition.

© ISO 2019 – All rights reserved

19.4.13 classification_role

A **classification_role** is a specified purpose for including product data in an exchange and a description of that purpose.

EXPRESS specification:

```
ENTITY classification_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

19.4.14 contract_assignment

A **contract_assignment** is an association of a contract 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;
```

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.

Attribute definitions:

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.

19.4.15 date_and_time_assignment

A **date_and_time_assignment** is an association of a date_and_time with product data.

 © ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY date_and_time_assignment
  ABSTRACT SUPERTYPE;
  assigned_date_and_time : date_and_time;
  role : date_time_role;
END_ENTITY;
```

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.

19.4.16 date_assignment

A **date_assignment** is an association of a date with product data.

EXPRESS specification:

```
ENTITY date_assignment
  ABSTRACT SUPERTYPE;
  assigned_date : date;
  role : date_role;
END_ENTITY;
```

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**.

19.4.17 document_reference

A **document_reference** is an association of a document with product data.

EXPRESS specification:

```
ENTITY document_reference
  ABSTRACT SUPERTYPE;
  assigned_document : document;
  source : label;
DERIVE
  role : object_role := get_role(SELF);
WHERE
```

© ISO 2019 – All rights reserved

```
  WR1: SIZEOF(USEDIN(SELF, 'BASIC_ATTRIBUTE_SCHEMA.' + 'ROLE_ASSOCIATION.ITEM_WITH_
ROLE')) <= 1;
END_ENTITY;
```

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** with product data.

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.

Attribute definitions:

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.

19.4.18 document_usage_constraint_assignment

A **document_usage_constraint_assignment** is an association of a document_usage_constraint with product data.

EXPRESS specification:

```
ENTITY document_usage_constraint_assignment
  ABSTRACT SUPERTYPE;
  assigned_document_usage : document_usage_constraint;
  role : document_usage_role;
END_ENTITY;
```

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.

© ISO 2019 – All rights reserved

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.

19.4.19 document_usage_role

A **document_usage_role** is a specified purpose for including product data in an exchange and a description of that purpose.

EXPRESS specification:

```
ENTITY document_usage_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

19.4.20 effectivity_assignment

An **effectivity_assignment** is an association of an effectivity 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;
```

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.

Attribute definitions:

© ISO 2019 – All rights reserved

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.

19.4.21 effectivity_context_assignment

An **effectivity_context_assignment** is an assignment of a purpose to an effectivity_assignment.

EXPRESS specification:

```
ENTITY effectivity_context_assignment
  ABSTRACT SUPERTYPE;
  assigned_effectivity_assignment : effectivity_assignment;
  role : effectivity_context_role;
END_ENTITY;
```

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.

19.4.22 effectivity_context_role

An **effectivity_context_role** is a specified purpose and the product data that is the context for an effectivity_context_assignment and a description of that purpose.

EXPRESS specification:

```
ENTITY effectivity_context_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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 this attribute need not be specified.

19.4.23 event_occurrence_assignment

An **event_occurrence_assignment** is an association of an event_occurrence with product data.

 © ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY event_occurrence_assignment
  ABSTRACT SUPERTYPE;
  assigned_event_occurrence : event_occurrence;
  role : event_occurrence_role;
END_ENTITY;
```

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 association of the **event_occurrence_assignment** with product data.

19.4.24 event_occurrence_context_assignment

An **event_occurrence_context_assignment** is an identification of 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;
```

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.

19.4.25 experience_assignment

An **experience_assignment** is a mechanism for associating an experience with product data.

EXAMPLE The association of a particular episode of flying practice with a particular pilot.

EXPRESS specification:

```
ENTITY experience_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_experience : experience;
  role : experience_role;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience_assignment**.

© ISO 2019 – All rights reserved

name the label by which the **experience_assignment** is known.

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

assigned_experience the instance of the experience entity data type that is to be associated with product data.

role the experience_role that specifies the purpose of the association of the **experience_assignment** with product data.

19.4.26 experience_role

An **experience_role** is a specified purpose for an experience and the description of the purpose.

EXPRESS specification:

```
ENTITY experience_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience_role**.

name the label by which the **experience_role** is known.

description the text that characterizes the **experience_role**. The value of this attribute need not be specified.

19.4.27 experience_type_assignment

An **experience_type_assignment** is a mechanism for associating an experience_type with product data.

EXAMPLE The association of flying experience with a particular category of naval personnel.

EXPRESS specification:

```
ENTITY experience_type_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_experience_type : experience_type;
  role : experience_type_role;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience_type_assignment**.

name the label by which the **experience_type_assignment** is known.

description the text that characterizes the **experience_type_assignment**. The value of this attribute need not be specified.

assigned_experience_type the instance of the experience_type entity data type that is to be associated with product data.

 © ISO 2019 – All rights reserved

role the experience_type_role that specifies the purpose of the association of the **experience_type_assignment** with product data.

19.4.28 experience_type_role

An **experience_type_role** is a specified purpose for an experience_type and a description of the purpose.

EXPRESS specification:

```
ENTITY experience_type_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **experience_type_role**.

name the label by which the **experience_type_role** is known.

description the text that characterizes the **experience_type_role**. The value of this attribute need not be specified.

19.4.29 external_identification_assignment

An **external_identification_assignment** is a type of identification_assignment that is defined in the context of an external_source. 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;
```

Attribute definitions:

source the external_source that specifies the context in which in the source_id is defined.

19.4.30 external_identification_assignment_relationship

An **external_identification_assignment_relationship** is a relationship between two external_identification_assignment objects.

EXPRESS specification:

```
ENTITY external_identification_assignment_relationship;
  id : OPTIONAL identifier;
  description : OPTIONAL text;
  relating : external_identification_assignment;
  related : external_identification_assignment;
  relation_type : STRING;
WHERE
  WR1: acyclic_external_identification_assignment_relationship(SELF, [related],
'MANAGEMENT_RESOURCES_SCHEMA.EXTERNAL_IDENTIFICATION_ASSIGNMENT_RELATIONSHIP');
END_ENTITY;
```

Attribute definitions:

© ISO 2019 – All rights reserved

id the identifier or set of identifiers for the **external_identification_assignment_relationship**.

description the text or the set of texts that provide further information about the **external_identification_assignment_relationship**. The value of this attribute need not be specified.

relating one object of external_identification_assignment that is part of the relationship.

related the other object of external_identification_assignment that is part of the relationship.

relation_type the meaning of the relationship.

Attribute definitions:

WR1 An **external_identification_assignment_relationship** shall not participate in its own definition.

19.4.31 external_referent_assignment

An **external_referent_assignment** is an identification of product data referenced in outside sources.

NOTE 1 This is an incompletely modeled concept and requires population of related entity data types to form a complete **external_referent_assignment**.

EXAMPLE 1 A URI for an externally defined tool is specified by assigning the value 'http://www.tool.com/mill.stp' to the name attribute of external_referent_assignment. The fact that the value is a URI is established by assigning a value of 'uri' to the role attribute of external_referent_assignment.

EXAMPLE 2 A URI fragment identifier for the bottom face of the externally defined tool is specified by assigning the value 'http://www.tool.com/mill.stp#bottom_face' to the name attribute of external_referent_assignment. The fact that the value is a URI

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;
```

Attribute definitions:

assigned_name the label by which the **external_referent_assignment** 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.

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

NOTE 3 This attribute is an enhancement to the definition of **external_referent_assignment** using a method that is upwardly compatible with ISO 10303-41:1994.

 © ISO 2019 – All rights reserved

Attribute definitions:

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

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

19.4.32 group_assignment

A **group_assignment** is an association of a group 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;
```

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.

Attribute definitions:

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.

19.4.33 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;
```

Attribute definitions:

assigned_id the identification associated with the product data.

© ISO 2019 – All rights reserved

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.

19.4.34 identification_assignment_relationship

An **identification_assignment_relationship** is a relationship between 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;
```

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.

 © ISO 2019 – All rights reserved

19.4.35 identification_role

An **identification_role** is a purpose for identifying an item of product data and includes identification and description of the purpose.

EXPRESS specification:

```
ENTITY identification_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

19.4.36 library_assignment

A **library_assignment** is a type of external_referent_assignment that assigns a library_context to product data.

EXPRESS specification:

```
ENTITY library_assignment
  ABSTRACT SUPERTYPE
  SUBTYPE OF (external_referent_assignment);
  frame_of_reference : library_context;
UNIQUE
  UR1: frame_of_reference;
END_ENTITY;
```

Attribute definitions:

frame_of_reference the library_context in which the **library_assignment** is defined.

19.4.37 location_assignment

A **location_assignment** is a mechanism for associating a location with product data.

EXAMPLE The association of a geographical location with an item of heavy lifting equipment.

EXPRESS specification:

```
ENTITY location_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_location : location;
  role : location_role;
END_ENTITY;
```

Attribute definitions:

© ISO 2019 – All rights reserved

id the identifier that distinguishes the **location_assignment**.

name the label by which the **location_assignment** is known.

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

assigned_location the instance of the location entity data type that is to be associated with product data.

role the location_role that specifies the purpose of the association of the **location_assignment** with product data.

19.4.38 location_representation_assignment

A **location_representation_assignment** is a mechanism for associating a location with its representation.

EXAMPLE The association of a location with its representation as a set of geographical co-ordinates.

EXPRESS specification:

```
ENTITY location_representation_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  represented_location : location;
  role : location_representation_role;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **location_representation_assignment**.

name the label by which the **location_representation_assignment** is known.

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

represented_location the instance of the location entity data type that is to be associated with product data.

role the location_representation_role that specifies the purpose of the association of the **location_representation_assignment** with product data.

19.4.39 location_representation_role

A **location_representation_role** is a purpose for a representation of a location and provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY location_representation_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **location_representation_role**.

© ISO 2019 – All rights reserved

name the label by which the **location_representation_role** is known.

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

19.4.40 location_role

A **location_role** is the purpose for a location in the context of product data an provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY location_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **location_role**.

name the label by which the **location_role** is known.

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

19.4.41 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;
```

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

Attribute definitions:

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.

© ISO 2019 – All rights reserved

19.4.42 organization_assignment

An **organization_assignment** is an association of an organization with product data.

EXPRESS specification:

```
ENTITY organization_assignment
  ABSTRACT SUPERTYPE;
  assigned_organization : organization;
  role : organization_role;
END_ENTITY;
```

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**.

19.4.43 organization_type_assignment

An **organization_type_assignment** is a mechanism for associating an organization_type with product data.

EXAMPLE An association between national standards bodies and ISO.

EXPRESS specification:

```
ENTITY organization_type_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_organization_type : organization_type;
  role : organization_type_role;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **organization_type_assignment**.

name the label by which the **organization_type_assignment** is known.

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

assigned_organization_type the instance of the organization_type entity data type that is to be associated with product data.

role the organization_type_role that specifies the purpose of the association of the **organization_type_assignment** with product data.

19.4.44 organization_type_role

An **organization_type_role** is a purpose for an **organization_type** and provides identification and description of the purpose.

 © ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY organization_type_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **organization_type_role**.

name the label by which the **organization_type_role** is known.

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

19.4.45 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;
```

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.

19.4.46 organizational_project_role

An **organizational_project_role** is a purpose for an **organizational_project** and provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY organizational_project_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

name the label by which the **organizational_project_role** is known.

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

© ISO 2019 – All rights reserved

19.4.47 person_and_organization_assignment

A **person_and_organization_assignment** is an association of a person_and_organization 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;
```

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.

19.4.48 person_assignment

A **person_assignment** is the association of a person with product data.

EXPRESS specification:

```
ENTITY person_assignment
  ABSTRACT SUPERTYPE;
  assigned_person : person;
  role : person_role;
END_ENTITY;
```

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**.

19.4.49 person_type_assignment

A **person_type_assignment** is a mechanism for associating a person_type with product data.

EXAMPLE An association between a (generic) *structural engineer* and a particular construction project.

EXPRESS specification:

```
ENTITY person_type_assignment
  ABSTRACT SUPERTYPE;
```


© ISO 2019 – All rights reserved

```
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_person_type : person_type;
  role : person_type_role;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type_assignment**.

name the label by which the **person_type_assignment** is known.

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

assigned_person_type the instance of the person_type entity data type that is to be associated with product data.

role the person_type_role that specifies the purpose of the association of the **person_type_assignment** with product data.

19.4.50 person_type_definition_assignment

A **person_type_definition_assignment** is a mechanism for associating a person_type_definition with product data.

EXAMPLE An association between the definition of *structural engineer* employed by the UK Institution of Structural Engineers and the definition of *structural engineer* employed by a particular construction project.

EXPRESS specification:

```
ENTITY person_type_definition_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_person_type_definition : person_type_definition;
  role : person_type_definition_role;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type_definition_assignment**.

name the label by which the **person_type_definition_assignment** is known.

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

assigned_person_type_definition the instance of the person_type_definition entity data type that is to be associated with product data.

role the person_type_definition_role that specifies the purpose of the association of the **person_type_definition_assignment** with product data.

© ISO 2019 – All rights reserved

19.4.51 person_type_definition_role

A **person_type_definition_role** is a purpose for a **person_type_definition** and provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY person_type_definition_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type_definition_role**.

name the label by which the **person_type_definition_role** is known.

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

19.4.52 person_type_role

A **person_type_role** is a purpose for a **person_type** and provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY person_type_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type_role**.

name the label by which the **person_type_role** is known.

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

19.4.53 position_in_organization_assignment

A **position_in_organization_assignment** is a mechanism for associating a position_in_organization with product data.

EXAMPLE An association between the managing director of a particular organization and a project.

EXPRESS specification:

```
ENTITY position_in_organization_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_position_in_organization : position_in_organization;
  role : position_in_organization_role;
END_ENTITY;
```


© ISO 2019 – All rights reserved

Attribute definitions:

id the identifier that distinguishes the **position_in_organization_assignment**.

name the label by which the **position_in_organization_assignment** is known.

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

assigned_position_in_organization the instance of the position_in_organization entity data type that is to be associated with product data.

role the position_in_organization_role that specifies the purpose of the association of the **position_in_organization_assignment** with product data.

19.4.54 position_in_organization_role

A **position_in_organization_role** is a purpose for a **position_in_organization** and provides identification and description of the purpose

EXPRESS specification:

```
ENTITY position_in_organization_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **position_in_organization_role**.

name the label by which the **position_in_organization_role** is known.

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

19.4.55 position_in_organization_type_assignment

A **position_in_organization_type_assignment** is a mechanism for associating a position_in_organization_type with product data.

EXAMPLE An association between a generic company manager and a project.

EXPRESS specification:

```
ENTITY position_in_organization_type_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_position_in_organization_type : position_in_organization_type;
  role : position_in_organization_type_role;
END_ENTITY;
```

Attribute definitions:

© ISO 2019 – All rights reserved

id the identifier that distinguishes the **position_in_organization_type_assignment**.

name the label by which the **position_in_organization_type_assignment** is known.

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

assigned_position_in_organization_type the instance of the position_in_organization_type entity data type that is to be associated with product data.

role the position_in_organization_type_role that specifies the purpose of the association of the **position_in_organization_type_assignment** with product data.

19.4.56 position_in_organization_type_role

A **position_in_organization_type_role** is a purpose for a **position_in_organization_type** and provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY position_in_organization_type_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **position_in_organization_type_role**.

name the label by which the **position_in_organization_type_role** is known.

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

19.4.57 qualification_assignment

A **qualification_assignment** is a mechanism for associating a qualification with product data.

EXAMPLE The association of a pilot's licence with a person indicates that the person is qualified to fulfill the role of a pilot.

EXPRESS specification:

```
ENTITY qualification_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_qualification : qualification;
  role : qualification_role;
END_ENTITY;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

id the identifier that distinguishes the **qualification_assignment**.

name the label by which the **qualification_assignment** is known.

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

assigned_qualification the instance of the qualification entity data type that is to be associated with product data.

role the qualification_role that specifies the purpose of the association of the **qualification_assignment** with product data.

19.4.58 qualification_role

A **qualification_role** is a purpose for a qualification and provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY qualification_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **qualification_role**.

name the label by which the **qualification_role** is known.

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

19.4.59 qualification_type_assignment

A **qualification_type_assignment** is a mechanism for associating a qualification_type with product data.

EXAMPLE An association between a person and a Product Management Professional certification indicates that the person is qualified to take on the role of project manager.

EXPRESS specification:

```
ENTITY qualification_type_assignment
  ABSTRACT SUPERTYPE;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  assigned_qualification_type : qualification_type;
  role : qualification_type_role;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **qualification_type_assignment**.

name the label by which the **qualification_type_assignment** is known.

© ISO 2019 – All rights reserved

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

assigned_qualification_type the instance of the qualification_type entity data type that is to be associated with product data.

role the qualification_type_role that specifies the purpose of the association of the **qualification_type_assignment** with product data.

19.4.60 qualification_type_role

A **qualification_type_role** is a purpose for a **qualification_type** and provides identification and description of the purpose.

EXPRESS specification:

```
ENTITY qualification_type_role;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **qualification_type_role**.

name the label by which the **qualification_type_role** is known.

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

19.4.61 security_classification_assignment

A **security_classification_assignment** is an association of a security_classification 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;
```

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

Attribute definitions:

 © ISO 2019 – All rights reserved

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.

19.4.62 time_assignment

A **time_assignment** is an association of a local_time with product data.

EXPRESS specification:

```
ENTITY time_assignment
  ABSTRACT SUPERTYPE;
  assigned_time : local_time;
  role : time_role;
END_ENTITY;
```

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.

19.4.63 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;
```

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.

19.5 management_resources_schema function definitions

19.5.1 acyclic_assignment_object_relationship

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

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

© ISO 2019 – All rights reserved

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

EXPRESS specification:

```
FUNCTION acyclic_assignment_object_relationship(relation : assignment_object_relationship;
                                                relatives : SET [1:?] OF assignment_
object_select;
                                                specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF assignment_object_relationship;
  END_LOCAL;
  IF relation.relating IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(aor <* bag_to_set(USEDIN(relation.relating, ('MANAGEMENT_RESOURCES_SCHEMA.'
+ 'ASSIGNMENT_OBJECT_RELATIONSHIP.') + 'RELATED')) | specific_relation IN TYPEOF(aor));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_assignment_object_relationship(x[i], relatives + relation.relating,
specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

19.5.1.1 Parameters

19.5.1.1.1 relation

(input) the candidate assignment_object_relationship to be checked.

EXPRESS specification:

```
relation : assignment_object_relationship
```

19.5.1.1.2 relatives

(input) the set of instances of the entity data type assignment_object_select for which the function is searching in the relating parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF assignment_object_select
```

19.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the assignment_object_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

19.5.1.2 Variables

19.5.1.2.1 x

EXPRESS specification:

```
x : SET OF assignment_object_relationship;
```


© ISO 2019 – All rights reserved

19.5.1.3 Statements

19.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_assignment_object_relationship(x[i], relatives + relation.relating,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

19.5.2 acyclic_classification_assignment_relationship

The **acyclic_classification_assignment_relationship** function determines whether the graph of instances of the entity data type classification_assignment that contains relation as one of its links contains a cycle. This function may be used to evaluate either a classification_assignment_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_classification_assignment_relationship(relation : classification_
assignment_relationship;
                                                        relatives : SET [1:?] OF
classification_assignment;
                                                        specific_relation : STRING) :
BOOLEAN;
  LOCAL
    x : SET OF classification_assignment_relationship;
  END_LOCAL;
  IF relation.relating IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(car <* bag_to_set(USEDIN(relation.relating, ('MANAGEMENT_RESOURCES_SCHEMA.
' + 'CLASSIFICATION_ASSIGNMENT_RELATIONSHIP.') + 'RELATED')) | specific_relation IN
TYPEOF(car));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_classification_assignment_relationship(x[i], relatives + relation.
relating, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

19.5.2.1 Parameters

19.5.2.1.1 relation

(input) the candidate classification_assignment_relationship to be checked.

EXPRESS specification:

```
relation : classification_assignment_relationship
```

19.5.2.1.2 relatives

(input) the set of instances of the entity data type classification_assignment for which the function is searching in the relating parameter of the relation argument.

© ISO 2019 – All rights reserved

EXPRESS specification:

```
relatives : SET [1:?] OF classification_assignment
```

19.5.2.1.3 specific_relation

(input) the fully qualified name of a subtype of the classification_assignment_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

19.5.2.2 Variables

19.5.2.2.1 x

EXPRESS specification:

```
x : SET OF classification_assignment_relationship;
```

19.5.2.3 Statements

19.5.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_classification_assignment_relationship(x[i], relatives + relation.
relating, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

19.5.3 acyclic_external_identification_assignment_relationship

The **acyclic_external_identification_assignment_relationship** function determines whether the graph of instances of the entity data type external_identification_assignment that contains relation as one of its links contains a cycle. This function may be used to evaluate either a external_identification_assignment_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_external_identification_assignment_relationship(relation : external_
identification_assignment_relationship;
                                                                 relatives : SET [1:?] OF
external_identification_assignment;
                                                                 specific_relation :
STRING) : BOOLEAN;
  LOCAL
    x : SET OF external_identification_assignment_relationship;
  END_LOCAL;
  IF relation.relating IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(eiar <* bag_to_set(USEDIN(relation.relating, ('MANAGEMENT_RESOURCES_SCHEMA.'
+ 'EXTERNAL_IDENTIFICATION_ASSIGNMENT_RELATIONSHIP.') + 'RELATED')) | specific_relation
IN TYPEOF(eiar));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_external_identification_assignment_relationship(x[i], relatives +
relation.relating, specific_relation) THEN
      RETURN (FALSE);
```

 © ISO 2019 – All rights reserved

```
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

19.5.3.1 Parameters

19.5.3.1.1 relation

(input) the candidate external_identification_assignment_relationship to be checked.

EXPRESS specification:

```
relation : external_identification_assignment_relationship
```

19.5.3.1.2 relatives

(input) the set of instances of the entity data type external_identification_assignment for which the function is searching in the relating parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF external_identification_assignment
```

19.5.3.1.3 specific_relation

(input) the fully qualified name of a subtype of the external_identification_assignment_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

19.5.3.2 Variables

19.5.3.2.1 x

EXPRESS specification:

```
x : SET OF external_identification_assignment_relationship;
```

19.5.3.3 Statements

19.5.3.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_external_identification_assignment_relationship(x[i], relatives +
relation.relating, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

19.5.4 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 an 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.

© ISO 2019 – All rights reserved

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.relating_identification_assignment IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(ia <* bag_to_set(USEDIN(relation.relating_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.
relating_identification_assignment, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

19.5.4.1 Parameters

19.5.4.1.1 relation

(input) the candidate identification_assignment_relationship to be checked.

EXPRESS specification:

```
relation : identification_assignment_relationship
```

19.5.4.1.2 relatives

(input) the set of instances of the entity data type identification_assignment that the function is searching for in the relating_identification_assignment parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF identification_assignment
```

19.5.4.1.3 specific_relation

(input) the fully qualified entity name of a type of identification_assignment_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

19.5.4.2 Variables

19.5.4.2.1 x

EXPRESS specification:

```
x : SET OF identification_assignment_relationship;
```

 © ISO 2019 – All rights reserved

19.5.4.3 Statements

19.5.4.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_identification_assignment_relationship(x[i], relatives + relation.
relating_identification_assignment, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

20 measure_schema

20.1 General

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.

EXPRESS specification:

```
SCHEMA measure_schema '{iso standard 10303 part(41) version(8) object(1) measure_
schema(16)}';

REFERENCE FROM basic_attribute_schema
  (get_name_value,
   name_attribute,
   name_attribute_select);

REFERENCE FROM representation_schema
  (representation_context);

REFERENCE FROM support_resource_schema;
```

20.2 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.

20.3 measure_schema type definitions

20.3.1 s_name_attribute_select

EXPRESS specification:

```
TYPE s_name_attribute_select = SELECT BASED_ON name_attribute_select WITH
  (derived_unit);
END_TYPE;
```

© ISO 2019 – All rights reserved

20.3.2 absorbed_dose_measure

An **absorbed_dose_measure** is the value of an absorbed dose of radiation.

EXPRESS specification:

```
TYPE absorbed_dose_measure = REAL;
END_TYPE;
```

20.3.3 acceleration_measure

An **acceleration_measure** is the value of the rate of change of velocity.

EXPRESS specification:

```
TYPE acceleration_measure = REAL;
END_TYPE;
```

20.3.4 radioactivity_measure

A **radioactivity_measure** is the value of the radioactive disintegration.

EXPRESS specification:

```
TYPE radioactivity_measure = REAL;
END_TYPE;
```

20.3.5 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;
```

20.3.6 area_measure

An **area_measure** is the value of the extent of a surface.

EXPRESS specification:

```
TYPE area_measure = REAL;
END_TYPE;
```

20.3.7 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;
```

20.3.8 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;
```

 © ISO 2019 – All rights reserved

```
END_TYPE;
```

20.3.9 count_measure

A **count_measure** is the value of a count.

EXPRESS specification:

```
TYPE count_measure = NUMBER;
END_TYPE;
```

20.3.10 descriptive_measure

A **descriptive_measure** is a textual value of a physical quantity.

EXPRESS specification:

```
TYPE descriptive_measure = STRING;
END_TYPE;
```

20.3.11 capacitance_measure

A **capacitance_measure** is the value of capacitance.

EXPRESS specification:

```
TYPE capacitance_measure = REAL;
END_TYPE;
```

20.3.12 dose_equivalent_measure

A **dose_equivalent_measure** is the value of the radiation dose equivalent.

EXPRESS specification:

```
TYPE dose_equivalent_measure = REAL;
END_TYPE;
```

20.3.13 electric_charge_measure

An **electric_charge_measure** is the value of an electrical charge.

EXPRESS specification:

```
TYPE electric_charge_measure = REAL;
END_TYPE;
```

20.3.14 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;
```

20.3.15 conductance_measure

A **conductance_measure** is the value of an electrical conductance.

EXPRESS specification:

```
TYPE conductance_measure = REAL;
```

© ISO 2019 – All rights reserved

```
END_TYPE;
```

20.3.16 electric_potential_measure

An **electric_potential_measure** is the value of an electrical potential.

EXPRESS specification:

```
TYPE electric_potential_measure = REAL;
END_TYPE;
```

20.3.17 energy_measure

An **energy_measure** is the value of energy, or work done, in a system.

EXPRESS specification:

```
TYPE energy_measure = REAL;
END_TYPE;
```

20.3.18 force_measure

A **force_measure** is the value of a force.

EXPRESS specification:

```
TYPE force_measure = REAL;
END_TYPE;
```

20.3.19 frequency_measure

A **frequency_measure** is the value of a frequency.

EXPRESS specification:

```
TYPE frequency_measure = REAL;
END_TYPE;
```

20.3.20 magnetic_flux_density_measure

A **magnetic_flux_density_measure** is the value of magnetic flux density.

EXPRESS specification:

```
TYPE magnetic_flux_density_measure = REAL;
END_TYPE;
```

20.3.21 illuminance_measure

An **illuminance_measure** is the value of illuminance.

EXPRESS specification:

```
TYPE illuminance_measure = REAL;
END_TYPE;
```

20.3.22 inductance_measure

An **inductance_measure** is the value of inductance.

EXPRESS specification:

```
TYPE inductance_measure = REAL;
```

 © ISO 2019 – All rights reserved

```
END_TYPE;
```

20.3.23 length_measure

A **length_measure** is the value of a distance.

EXPRESS specification:

```
TYPE length_measure = REAL;
END_TYPE;
```

20.3.24 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;
```

20.3.25 luminous_flux_measure

A **luminous_flux_measure** is the value of luminous flux.

EXPRESS specification:

```
TYPE luminous_flux_measure = REAL;
END_TYPE;
```

20.3.26 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;
```

20.3.27 measure_value

A **measure_value** is a value as defined in ISO 31-0.

EXPRESS specification:

```
TYPE measure_value = SELECT
  (absorbed_dose_measure,
   dose_equivalent_measure,
   radioactivity_measure,
   acceleration_measure,
   amount_of_substance_measure,
   area_measure,
   celsius_temperature_measure,
   context_dependent_measure,
   count_measure,
   descriptive_measure,
   capacitance_measure,
   electric_charge_measure,
   conductance_measure,
   electric_current_measure,
   electric_potential_measure,
   energy_measure,
   magnetic_flux_density_measure,
   force_measure,
   frequency_measure,
   illuminance_measure,
```

© ISO 2019 – All rights reserved

```
   inductance_measure,
   length_measure,
   luminous_flux_measure,
   luminous_intensity_measure,
   magnetic_flux_measure,
   mass_measure,
   numeric_measure,
   non_negative_length_measure,
   parameter_value,
   plane_angle_measure,
   positive_length_measure,
   positive_plane_angle_measure,
   positive_ratio_measure,
   power_measure,
   pressure_measure,
   ratio_measure,
   resistance_measure,
   solid_angle_measure,
   thermodynamic_temperature_measure,
   time_measure,
   velocity_measure,
   volume_measure);
END_TYPE;
```

20.3.28 magnetic_flux_measure

A **magnetic_flux_measure** is the value of magnetic flux.

EXPRESS specification:

```
TYPE magnetic_flux_measure = REAL;
END_TYPE;
```

20.3.29 non_negative_length_measure

A **non_negative_length_measure** type is a length_measure whose value is greater than or equal to zero.

EXPRESS specification:

```
TYPE non_negative_length_measure = length_measure;
WHERE
  WR1: SELF >= 0.0;
END_TYPE;
```

20.3.30 numeric_measure

A **numeric_measure** is the numeric value of a physical quantity.

NOTE In order to define measure quantities that are not included as specific types in this standard a **numeric_measure** may be used as the value_component of a measure_with_unit, the corresponding unit_component being an si_unit or a derived_unit.

EXPRESS specification:

```
TYPE numeric_measure = NUMBER;
END_TYPE;
```

20.3.31 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;
```

 © ISO 2019 – All rights reserved

```
END_TYPE;
```

20.3.32 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;
```

20.3.33 positive_length_measure

A **positive_length_measure** is a length_measure that is greater than zero.

EXPRESS specification:

```
TYPE positive_length_measure = non_negative_length_measure;
WHERE
  WR1: SELF > 0.0;
END_TYPE;
```

20.3.34 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;
```

20.3.35 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;
```

20.3.36 power_measure

A **power_measure** is the value of power, or the rate of doing work.

EXPRESS specification:

```
TYPE power_measure = REAL;
END_TYPE;
```

20.3.37 pressure_measure

A **pressure_measure** is the value of force per unit area.

EXPRESS specification:

```
TYPE pressure_measure = REAL;
END_TYPE;
```

© ISO 2019 – All rights reserved

20.3.38 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;
```

20.3.39 resistance_measure

A **resistance_measure** is the value of electrical resistance.

EXPRESS specification:

```
TYPE resistance_measure = REAL;
END_TYPE;
```

20.3.40 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;
```

20.3.41 si_unit_name

An **si_unit_name** is the name of an SI unit.

NOTE 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,
```

 © ISO 2019 – All rights reserved

```
   joule,
   watt,
   coulomb,
   volt,
   farad,
   ohm,
   siemens,
   weber,
   tesla,
   henry,
   degree_Celsius,
   lumen,
   lux,
   becquerel,
   gray,
   sievert);
END_TYPE;
```

20.3.42 solid_angle_measure

A **solid_angle_measure** is the value of a solid angle.

EXPRESS specification:

```
TYPE solid_angle_measure = REAL;
END_TYPE;
```

20.3.43 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;
```

20.3.44 time_measure

A **time_measure** is the value of the duration of a period.

EXPRESS specification:

```
TYPE time_measure = REAL;
END_TYPE;
```

20.3.45 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
  (derived_unit,
   named_unit);
END_TYPE;
```

20.3.46 velocity_measure

A **velocity_measure** is the value of the rate of change of position.

EXPRESS specification:

```
TYPE velocity_measure = REAL;
```

© ISO 2019 – All rights reserved

```
END_TYPE;
```

20.3.47 volume_measure

A **volume_measure** is the value of the solid content of a body.

EXPRESS specification:

```
TYPE volume_measure = REAL;
END_TYPE;
```

20.4 measure_schema entity definitions

20.4.1 absorbed_dose_measure_with_unit

An **absorbed_dose_measure_with_unit** is a type of measure_with_unit in which the physical quantity is an absorbed dose of radiation as defined in ISO 31.

EXPRESS specification:

```
ENTITY absorbed_dose_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.ABSORBED_DOSE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component be of type absorbed_dose_unit.

20.4.2 absorbed_dose_unit

An **absorbed_dose_unit** is a type of derived_unit in which the absorbed dose is expressed.

EXPRESS specification:

```
ENTITY absorbed_dose_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.gray);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the Gray as defined in ISO 31.

20.4.3 si_absorbed_dose_unit

A **si_absorbed_dose_unit** is a type of absorbed_dose_unit and a type of si_unit in which the absorbed dose is expressed in SI units.

EXPRESS specification:

```
ENTITY si_absorbed_dose_unit
  SUBTYPE OF (absorbed_dose_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.gray;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```


© ISO 2019 – All rights reserved

Attribute definitions:

WR1 The name of the si_unit shall be gray.

WR2 No other name shall be assigned to this entity.

20.4.4 acceleration_measure_with_unit

An **acceleration_measure_with_unit** is a type of measure_with_unit in which the physical quantity is an acceleration.

EXPRESS specification:

```
ENTITY acceleration_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.ACCELERATION_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component be of type acceleration_unit.

20.4.5 acceleration_unit

An **acceleration_unit** is a type of derived_unit in which the acceleration of an object is expressed.

EXPRESS specification:

```
ENTITY acceleration_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensional_exponents(1.0, 0.0, -2.0, 0.0,
0.0, 0.0, 0.0);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponent of length shall be equal to one, the dimensional exponent of time shall be equal to minus two, and all the other dimensional exponents shall be equal to zero.

20.4.6 radioactivity_measure_with_unit

A **radioactivity_measure_with_unit** is a type of measure_with_unit in which the physical quantity is radioactivity as defined in ISO 31.

EXPRESS specification:

```
ENTITY radioactivity_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.RADIOACTIVITY_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type radioactivity_unit.

© ISO 2019 – All rights reserved

20.4.7 radioactivity_unit

A **radioactivity_unit** is a type of derived_unit in which the radioactivity is expressed.

EXPRESS specification:

```
ENTITY radioactivity_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.
becquerel);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the becquerel as defined in ISO 31.

20.4.8 si_radioactivity_unit

A **si_radioactivity_unit** is a type of radioactivity_unit and a type of si_unit in which the radioactivity is expressed in SI units.

EXPRESS specification:

```
ENTITY si_radioactivity_unit
  SUBTYPE OF (radioactivity_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.becquerel;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be becquerel.

WR2 No other name shall be assigned to this entity.

20.4.9 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 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;
```

Attribute definitions:

WR1 The unit shall be an amount_of_substance_unit.

20.4.10 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.

 © ISO 2019 – All rights reserved

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 (SELF\named_unit.dimensions.luminous_intensity_
exponent = 0.0);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponent of amount of substance shall be equal to one, and all the other dimensional exponents shall be equal to zero.

20.4.11 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\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit shall be an area_unit.

20.4.12 area_unit

An **area_unit** is a type of derived_unit used to measure the extent of a surface.

EXPRESS specification:

```
ENTITY area_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensional_exponents(2.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponent of length shall be equal to two and all the other dimensional exponents shall be equal to zero.

© ISO 2019 – All rights reserved

20.4.13 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;
```

Attribute definitions:

WR1 The unit shall be a thermodynamic_temperature_unit.

20.4.14 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;
```

Attribute definitions:

name the label by which the **context_dependent_unit** is known.

20.4.15 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;
WHERE
  WR1: SELF\named_unit.dimensions = derive_dimensional_exponents(conversion_factor
\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

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.

Attribute definitions:

WR1 The dimensional exponents shall be equal to those from the conversion_factor.

20.4.16 derived_unit

A **derived_unit** is an expression of units.

EXAMPLE Newtons per square millimetre is a **derived_unit**.

A **derived_unit** may be either an absorbed_dose_unit or an acceleration_unit, or a radioactivity_unit, or an area_unit, or a capacitance_unit, or a dose_equivalent_unit, or an electric_charge_unit, or a conductance_unit, or an electric_potential_unit, or an energy_unit, or a magnetic_flux_density_unit or a force_unit, or a frequency_unit, or an illuminance_unit, or an inductance_unit, or a magnetic_flux_unit, or a power_unit, or a pressure_unit, or a resistance_unit, or a velocity_unit, or a volume_unit.

EXPRESS specification:

```
ENTITY derived_unit
  SUPERTYPE OF (ONEOF(absorbed_dose_unit, acceleration_unit, radioactivity_unit, area_
unit, capacitance_unit, dose_equivalent_unit, electric_charge_unit, conductance_unit,
electric_potential_unit, energy_unit, magnetic_flux_density_unit, force_unit, frequency_
unit, illuminance_unit, inductance_unit, magnetic_flux_unit, power_unit, pressure_unit,
resistance_unit, velocity_unit, volume_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;
```

Attribute definitions:

elements the derived_unit_elements 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.

Attribute definitions:

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 basic_attribute_schema 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.

© ISO 2019 – All rights reserved

20.4.17 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;
```

Attribute definitions:

unit the named_unit that specifies the mathematical factor of the element.

exponent the power that is applied to the unit attribute.

20.4.18 dimensional_exponents

A **dimensional_exponents** is a definition of 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;
```

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.

© ISO 2019 – All rights reserved

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.

20.4.19 capacitance_unit

A **capacitance_unit** is a type of derived_unit in which the capacitance is expressed.

EXPRESS specification:

```
ENTITY capacitance_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.farad);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the farad as defined in ISO 31.

20.4.20 capacitance_measure_with_unit

A **capacitance_measure_with_unit** is a type of measure_with_unit in which the physical quantity is an electric capacitance as defined in ISO 31.

EXPRESS specification:

```
ENTITY capacitance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.CAPACITANCE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component be of type capacitance_unit.

20.4.21 si_capacitance_unit

A **si_capacitance_unit** is a type of capacitance_unit and a type of si_unit in which the electric capacitance is expressed in SI units.

EXPRESS specification:

```
ENTITY si_capacitance_unit
  SUBTYPE OF (capacitance_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.farad;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be farad.

WR2 No other name shall be assigned to this entity.

© ISO 2019 – All rights reserved

20.4.22 dose_equivalent_measure_with_unit

A **dose_equivalent_measure_with_unit** is a type of measure_with_unit in which the physical quantity is a radiation dose equivalent as defined in ISO 31.

EXPRESS specification:

```
ENTITY dose_equivalent_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.DOSE_EQUIVALENT_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type dose_equivalent_unit.

20.4.23 dose_equivalent_unit

A **dose_equivalent_unit** is a type of derived_unit in which the radiation dose equivalent is expressed.

EXPRESS specification:

```
ENTITY dose_equivalent_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.sievert);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the sievert as defined in ISO 31.

20.4.24 si_dose_equivalent_unit

A **si_dose_equivalent_unit** is a type of dose_equivalent_unit and a type of si_unit in which the radiation dose equivalent is expressed in SI units.

EXPRESS specification:

```
ENTITY si_dose_equivalent_unit
  SUBTYPE OF (dose_equivalent_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.sievert;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be sievert.

WR2 No other name shall be assigned to this entity.

20.4.25 electric_charge_unit

An **electric_charge_unit** is a type of derived_unit in which the electric charge is expressed.

EXPRESS specification:

```
ENTITY electric_charge_unit
```


© ISO 2019 – All rights reserved

```
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.coulomb);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the coulomb as defined in ISO 31.

20.4.26 electric_charge_measure_with_unit

An **electric_charge_measure_with_unit** is a type of measure_with_unit in which the physical quantity is an electric charge as defined in ISO 31.

EXPRESS specification:

```
ENTITY electric_charge_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.ELECTRIC_CHARGE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type electric_charge_unit.

20.4.27 si_electric_charge_unit

A **si_electric_charge_unit** is a type of electric_charge_unit and a type of si_unit in which the electric charge is expressed in SI units.

EXPRESS specification:

```
ENTITY si_electric_charge_unit
  SUBTYPE OF (electric_charge_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.coulomb;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be volt.

WR2 No other name shall be assigned to this entity.

20.4.28 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;
```

© ISO 2019 – All rights reserved

Attribute definitions:

WR1 The unit shall be an electric_current_unit.

20.4.29 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;
```

Attribute definitions:

WR1 The dimensional exponent of electric current shall be equal to one, and all the other dimensional exponents shall be equal to zero.

20.4.30 conductance_unit

A **conductance_unit** is a type of derived_unit in which the electrical conductance is expressed.

EXPRESS specification:

```
ENTITY conductance_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.siemens);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the siemens as defined in ISO 31.

20.4.31 conductance_measure_with_unit

A **conductance_measure_with_unit** is a type of measure_with_unit in which the physical quantity is a conductance as defined in ISO 31.

EXPRESS specification:

```
ENTITY conductance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.CONDUCTANCE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

WR1 The unit_component shall be of type conductance_unit.

20.4.32 si_conductance_unit

A **si_conductance_unit** is a type of conductance_unit and a type of si_unit in which the electrical conductance is expressed in SI units.

EXPRESS specification:

```
ENTITY si_conductance_unit
  SUBTYPE OF (conductance_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.siemens;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be siemens.

WR2 No other name shall be assigned to this entity.

20.4.33 electric_potential_unit

An **electric_potential_unit** is a type of derived_unit in which the electric potential is expressed.

EXPRESS specification:

```
ENTITY electric_potential_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.volt);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the volt as defined in ISO 31.

20.4.34 electric_potential_measure_with_unit

An **electric_potential_measure_with_unit** is a type of measure_with_unit in which the physical quantity is an electric potential as defined in ISO 31.

EXPRESS specification:

```
ENTITY electric_potential_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.ELECTRIC_POTENTIAL_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type electric_potential_unit.

© ISO 2019 – All rights reserved

20.4.35 si_electric_potential_unit

A **si_electric_potential_unit** is a type of electric_potential_unit and a type of si_unit in which the electric potential is expressed in SI units.

EXPRESS specification:

```
ENTITY si_electric_potential_unit
  SUBTYPE OF (electric_potential_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.volt;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be volt.

WR2 No other name shall be assigned to this entity.

20.4.36 energy_unit

An **energy_unit** is a type of derived_unit in which the energy is expressed.

EXPRESS specification:

```
ENTITY energy_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.joule);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the joule as defined in ISO 31.

20.4.37 energy_measure_with_unit

An **energy_measure_with_unit** is a type of measure_with_unit in which the physical quantity is energy as defined in ISO 31.

EXPRESS specification:

```
ENTITY energy_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.ENERGY_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type energy_unit.

20.4.38 si_energy_unit

A **si_energy_unit** is a type of energy_unit and a type of si_unit in which the energy is expressed in SI units.

EXPRESS specification:

```
ENTITY si_energy_unit
  SUBTYPE OF (energy_unit, si_unit);
```


© ISO 2019 – All rights reserved

```
WHERE
  WR1: SELF\si_unit.name = si_unit_name.joule;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be joule.

WR2 No other name shall be assigned to this entity.

20.4.39 magnetic_flux_density_unit

A **magnetic_flux_density_unit** is a type of derived_unit in which the magnetic flux density is expressed.

EXPRESS specification:

```
ENTITY magnetic_flux_density_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.tesla);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the tesla as defined in ISO 31.

20.4.40 magnetic_flux_density_measure_with_unit

A **magnetic_flux_density_measure_with_unit** is a type of measure_with_unit in which the physical quantity is magnetic flux density as defined in ISO 31.

EXPRESS specification:

```
ENTITY magnetic_flux_density_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.MAGNETIC_FLUX_DENSITY_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type magnetic_flux_density_unit.

20.4.41 si_magnetic_flux_density_unit

A **si_magnetic_flux_density_unit** is a type of magnetic_flux_density_unit and a type of si_unit in which the magnetic flux density is expressed in SI units.

EXPRESS specification:

```
ENTITY si_magnetic_flux_density_unit
  SUBTYPE OF (magnetic_flux_density_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.tesla;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

© ISO 2019 – All rights reserved

WR1 The name of the si_unit shall be tesla.

WR2 No other name shall be assigned to this entity.

20.4.42 force_unit

A **force_unit** is a type of derived_unit in which the force is expressed.

EXPRESS specification:

```
ENTITY force_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.newton);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the newton as defined in ISO 31.

20.4.43 force_measure_with_unit

A **force_measure_with_unit** is a type of measure_with_unit in which the physical quantity is force as defined in ISO 31.

EXPRESS specification:

```
ENTITY force_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.FORCE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type force_unit.

20.4.44 si_force_unit

A **si_force_unit** is a type of force_unit and a type of si_unit in which the force is expressed in SI units.

EXPRESS specification:

```
ENTITY si_force_unit
  SUBTYPE OF (force_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.newton;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be newton.

WR2 No other name shall be assigned to this entity.

© ISO 2019 – All rights reserved

20.4.45 frequency_unit

A **frequency_unit** is a type of derived_unit in which the frequency is expressed.

EXPRESS specification:

```
ENTITY frequency_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.hertz);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the hertz as defined in ISO 31.

20.4.46 frequency_measure_with_unit

A **frequency_measure_with_unit** is a type of measure_with_unit in which the quantity measured is a frequency as defined in ISO 31.

EXPRESS specification:

```
ENTITY frequency_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.FREQUENCY_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type frequency_unit.

20.4.47 si_frequency_unit

A **si_frequency_unit** is a type of frequency_unit and a type of si_unit in which the frequency is expressed in SI units.

EXPRESS specification:

```
ENTITY si_frequency_unit
  SUBTYPE OF (frequency_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.hertz;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be hertz.

WR2 No other name shall be assigned to this entity.

20.4.48 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.

NOTE For an example of the use of this entity, see annex F.4.6.

© ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY global_unit_assigned_context
  SUBTYPE OF (representation_context);
  units : SET [1:?] OF unit;
END_ENTITY;
```

Attribute definitions:

units the units that apply in the representation_context.

20.4.49 illuminance_unit

An **illuminance_unit** is a type of derived_unit in which the illuminance is expressed.

EXPRESS specification:

```
ENTITY illuminance_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.lux);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the lux as defined in ISO 31.

20.4.50 illuminance_measure_with_unit

An **illuminance_measure_with_unit** is a type of measure_with_unit in which the quantity measured is an illuminance as defined in ISO 31.

EXPRESS specification:

```
ENTITY illuminance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.ILLUMINANCE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type illuminance_unit.

20.4.51 si_illuminance_unit

A **si_illuminance_unit** is a type of illuminance_unit and a type of si_unit in which the illuminance is expressed in SI units.

EXPRESS specification:

```
ENTITY si_illuminance_unit
  SUBTYPE OF (illuminance_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.lux;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

WR1 The name of the si_unit shall be lux.

WR2 No other name shall be assigned to this entity.

20.4.52 inductance_unit

An **inductance_unit** is a type of derived_unit in which the inductance is expressed.

EXPRESS specification:

```
ENTITY inductance_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.henry);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the henry as defined in ISO 31.

20.4.53 inductance_measure_with_unit

An **inductance_measure_with_unit** is a type of measure_with_unit in which the quantity measured is an inductance as defined in ISO 31.

EXPRESS specification:

```
ENTITY inductance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.INDUCTANCE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type inductance_unit.

20.4.54 si_inductance_unit

A **si_inductance_unit** is a type of inductance_unit and a type of si_unit in which the inductance is expressed in SI units.

EXPRESS specification:

```
ENTITY si_inductance_unit
  SUBTYPE OF (inductance_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.henry;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be henry.

WR2 No other name shall be assigned to this entity.

© ISO 2019 – All rights reserved

20.4.55 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;
```

Attribute definitions:

WR1 The unit shall be a length_unit.

20.4.56 positive_length_measure_with_unit

A **positive_length_measure_with_unit** is a type of length_measure_with_unit of which the value is positive.

EXPRESS specification:

```
ENTITY positive_length_measure_with_unit
  SUBTYPE OF (length_measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.POSITIVE_LENGTH_MEASURE' IN TYPEOF(SELF\measure_with_unit.value_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The value of the value_component shall be positive.

20.4.57 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;
```

Attribute definitions:

WR1 The dimensional exponent of length shall be equal to one, and all the other dimensional exponents shall be equal to zero.

 © ISO 2019 – All rights reserved

20.4.58 luminous_flux_unit

A **luminous_flux_unit** is a type of named_unit in which the luminous flux is expressed.

EXPRESS specification:

```
ENTITY luminous_flux_unit
  SUBTYPE OF (named_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.lumen);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the lumen as defined in ISO 31.

20.4.59 luminous_flux_measure_with_unit

A **luminous_flux_measure_with_unit** is a type of measure_with_unit in which the quantity measured is a luminous flux as defined in ISO 31.

EXPRESS specification:

```
ENTITY luminous_flux_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.LUMINOUS_FLUX_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type luminous_flux_unit.

20.4.60 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;
```

Attribute definitions:

WR1 The unit shall be a luminous_intensity_unit.

20.4.61 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
```

© ISO 2019 – All rights reserved

```
  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;
```

Attribute definitions:

WR1 The dimensional exponent of luminous intensity shall be equal to one, and all the other dimensional exponents shall be equal to zero.

20.4.62 magnetic_flux_unit

A **magnetic_flux_unit** is a type of derived_unit in which the magnetic flux is expressed.

EXPRESS specification:

```
ENTITY magnetic_flux_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.weber);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the weber as defined in ISO 31.

20.4.63 magnetic_flux_measure_with_unit

A **magnetic_flux_measure_with_unit** is a type of measure_with_unit in which the quantity measured is a magnetic flux as defined in ISO 31.

EXPRESS specification:

```
ENTITY magnetic_flux_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.MAGNETIC_FLUX_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_
component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type magnetic_flux_unit.

20.4.64 si_magnetic_flux_unit

A **si_magnetic_flux_unit** is a type of magnetic_flux_unit and a type of si_unit in which the magnetic flux is expressed in SI units.

EXPRESS specification:

```
ENTITY si_magnetic_flux_unit
  SUBTYPE OF (magnetic_flux_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.weber;
  WR2: NOT EXISTS(SELF\derived_unit.name);
```

 © ISO 2019 – All rights reserved

```
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be weber.

WR2 No other name shall be assigned to this entity.

20.4.65 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;
```

Attribute definitions:

WR1 The unit shall be a mass_unit.

20.4.66 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;
```

Attribute definitions:

WR1 The dimensional exponent of mass shall be equal to one, and all the other dimensional exponents shall be equal to zero.

20.4.67 measure_with_unit

A **measure_with_unit** is the specification of a physical quantity as defined in ISO 31. A **measure_with_unit** may be either a length_measure_with_unit or a mass_measure_with_unit, or a time_measure_with_unit, or an electric_current_measure_with_unit, or a thermodynamic_temperature_measure_with_unit, or a celsius_temperature_measure_with_unit, or an amount_of_substance_measure_with_unit, or a luminous_intensity_measure_with_unit, or a plane_angle_measure_with_unit, or a solid_angle_measure_with_unit, or an area_measure_with_unit, or a volume_measure_with_unit, or a ratio_measure_with_unit, or an acceleration_measure_with_unit, or a capacitance_measure_with_unit, or an electric_charge_measure_with_unit, or

© ISO 2019 – All rights reserved

a conductance_measure_with_unit, or an electric_potential_measure_with_unit, or an energy_measure_with_unit, or a magnetic_flux_density_measure_with_unit, or a force_measure_with_unit, or a frequency_measure_with_unit, or an illuminance_measure_with_unit, or an inductance_measure_with_unit, or a luminous_flux_measure_with_unit, or a magnetic_flux_measure_with_unit, or a power_measure_with_unit, or a pressure_measure_with_unit, or a resistance_measure_with_unit, or a velocity_measure_with_unit, or an absorbed_dose_measure_with_unit, or a radioactivity_measure_with_unit, or a dose_equivalent_measure_with_unit.

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,
acceleration_measure_with_unit, capacitance_measure_with_unit, electric_charge_measure_
with_unit, conductance_measure_with_unit, electric_potential_measure_with_unit, energy_
measure_with_unit, magnetic_flux_density_measure_with_unit, force_measure_with_unit,
frequency_measure_with_unit, illuminance_measure_with_unit, inductance_measure_with_unit,
luminous_flux_measure_with_unit, magnetic_flux_measure_with_unit, power_measure_with_
unit, pressure_measure_with_unit, resistance_measure_with_unit, velocity_measure_with_
unit, absorbed_dose_measure_with_unit, radioactivity_measure_with_unit, dose_equivalent_
measure_with_unit));
  value_component : measure_value;
  unit_component : unit;
WHERE
  WR1: valid_units(SELF);
END_ENTITY;
```

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.

Attribute definitions:

WR1 The **unit_component** shall be a valid unit for the kind of measure specified by the **value_component**.

20.4.68 named_unit

A **named_unit** is a unit quantity. A **named_unit** may be either a si_unit or a conversion_based_unit, or a context_dependent_unit, and additionally may be a length_unit or a mass_unit, or a time_unit, or an electric_current_unit, or a thermodynamic_temperature_unit, or an amount_of_substance_unit, or a luminous_flux_unit, or a luminous_intensity_unit, or a plane_angle_unit, or a solid_angle_unit, or a ratio_unit.

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_flux_unit, luminous_intensity_unit, plane_angle_
unit, solid_angle_unit, ratio_unit));
  dimensions : dimensional_exponents;
END_ENTITY;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

dimensions the exponents of the base properties by which the **named_unit** is defined.

20.4.69 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;
```

Attribute definitions:

WR1 The unit shall be a plane_angle_unit.

20.4.70 positive_plane_angle_measure_with_unit

A **positive_plane_angle_measure_with_unit** is a type of plane_angle_measure_with_unit of which the value is positive.

EXPRESS specification:

```
ENTITY positive_plane_angle_measure_with_unit
  SUBTYPE OF (plane_angle_measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.POSITIVE_PLANE_ANGLE_MEASURE' IN TYPEOF(SELF\measure_with_unit.
value_component);
END_ENTITY;
```

Attribute definitions:

WR1 The value of the value_component shall be positive.

20.4.71 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;
```

Attribute definitions:

WR1 All the dimensional exponents shall be equal to zero.

© ISO 2019 – All rights reserved

20.4.72 power_unit

A **power_unit** is a type of derived_unit in which the power is expressed.

EXPRESS specification:

```
ENTITY power_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.watt);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the watt as defined in ISO 31.

20.4.73 power_measure_with_unit

A **power_measure_with_unit** is a type of measure_with_unit in which the quantity measured is a power as defined in ISO 31.

EXPRESS specification:

```
ENTITY power_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.POWER_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type power_unit.

20.4.74 si_power_unit

A **si_power_unit** is a type of power_unit and a type of si_unit in which the power is expressed in SI units.

EXPRESS specification:

```
ENTITY si_power_unit
  SUBTYPE OF (power_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.watt;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be watt.

WR2 No other name shall be assigned to this entity.

20.4.75 pressure_unit

A **pressure_unit** is a type of derived_unit in which the pressure is expressed.

EXPRESS specification:

```
ENTITY pressure_unit
  SUBTYPE OF (derived_unit);
WHERE
```

 © ISO 2019 – All rights reserved

```
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.pascal);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the pascal as defined in ISO 31.

20.4.76 pressure_measure_with_unit

A **pressure_measure_with_unit** is a type of measure_with_unit in which the quantity measured is a pressure as defined in ISO 31.

EXPRESS specification:

```
ENTITY pressure_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.PRESSURE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type pressure_unit.

20.4.77 si_pressure_unit

A **si_pressure_unit** is a type of pressure_unit and a type of si_unit in which the pressure is expressed in SI units.

EXPRESS specification:

```
ENTITY si_pressure_unit
  SUBTYPE OF (pressure_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.pascal;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be pascal.

WR2 No other name shall be assigned to this entity.

20.4.78 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;
```

Attribute definitions:

© ISO 2019 – All rights reserved

WR1 The unit shall be a ratio_unit.

20.4.79 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;
```

Attribute definitions:

WR1 All the dimensional exponents shall be equal to zero.

20.4.80 resistance_unit

A **resistance_unit** is a type of derived_unit in which the electrical resistance is expressed.

EXPRESS specification:

```
ENTITY resistance_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensions_for_si_unit(si_unit_name.ohm);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponents shall be equal to those of the ohm as defined in ISO 31.

20.4.81 resistance_measure_with_unit

A **resistance_measure_with_unit** is a type of measure_with_unit in which the quantity measured is an electrical resistance as defined in ISO 31.

EXPRESS specification:

```
ENTITY resistance_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
  WR1: 'MEASURE_SCHEMA.RESISTANCE_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type resistance_unit.

 © ISO 2019 – All rights reserved

20.4.82 si_resistance_unit

A **si_resistance_unit** is a type of resistance_unit and a type of si_unit in which the resistance is expressed in SI units.

EXPRESS specification:

```
ENTITY si_resistance_unit
  SUBTYPE OF (resistance_unit, si_unit);
WHERE
  WR1: SELF\si_unit.name = si_unit_name.ohm;
  WR2: NOT EXISTS(SELF\derived_unit.name);
END_ENTITY;
```

Attribute definitions:

WR1 The name of the si_unit shall be ohm.

WR2 No other name shall be assigned to this entity.

20.4.83 si_unit

An **si_unit** is a type of named_unit that defines a unit with respect to the system of units defined in ISO 80000.

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);
WHERE
  WR1: NOT (('MEASURE_SCHEMA.MASS_UNIT' IN TYPEOF(SELF)) AND (SIZEOF(USEDIN(SELF,
'MEASURE_SCHEMA.DERIVED_UNIT_ELEMENT.UNIT')) > 0)) OR (prefix = si_prefix.kilo);
END_ENTITY;
```

Attribute definitions:

prefix the si_prefix that specifies the ratio with the unit specified by name. The value of this attribute need not be specified.

name the si_unit_name by which the **si_unit** is known.

dimensions the dimensional_exponents for the **si_unit**.

Attribute definitions:

WR1 If the **si_unit** is a mass_unit it shall not be used by a derived_unit_element unless the prefix is kilo.

20.4.84 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\measure_with_unit.unit_component);
```

© ISO 2019 – All rights reserved

```
END_ENTITY;
```

Attribute definitions:

WR1 The unit shall be a solid_angle_unit.

20.4.85 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;
```

Attribute definitions:

WR1 All the dimensional exponents shall be equal to zero.

20.4.86 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;
```

Attribute definitions:

WR1 The unit shall be a thermodynamic_temperature_unit.

20.4.87 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.
```

 © ISO 2019 – All rights reserved

```
amount_of_substance_exponent = 0.0)) AND (SELF\named_unit.dimensions.luminous_intensity_
exponent = 0.0);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponent of thermodynamic temperature shall be equal to one and the other dimensional exponents shall be equal to zero.

20.4.88 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;
```

Attribute definitions:

WR1 The unit shall be a time_unit.

20.4.89 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;
```

Attribute definitions:

WR1 The dimensional exponent of time shall be equal to one, and all the other dimensional exponents shall be equal to zero.

20.4.90 velocity_measure_with_unit

A **velocity_measure_with_unit** is a type of measure_with_unit in which the quantity measured is a velocity.

EXPRESS specification:

```
ENTITY velocity_measure_with_unit
  SUBTYPE OF (measure_with_unit);
WHERE
```

© ISO 2019 – All rights reserved

```
  WR1: 'MEASURE_SCHEMA.VELOCITY_UNIT' IN TYPEOF(SELF\measure_with_unit.unit_component);
END_ENTITY;
```

Attribute definitions:

WR1 The unit_component shall be of type velocity_unit.

20.4.91 velocity_unit

A **velocity_unit** is a type of derived_unit in which the velocity is expressed.

EXPRESS specification:

```
ENTITY velocity_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensional_exponents(1.0, 0.0, -1.0, 0.0,
0.0, 0.0, 0.0);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponent of length shall be equal to one, the dimensional exponent of time shall be equal to minus one, and all the other dimensional exponents shall be equal to zero.

20.4.92 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;
```

Attribute definitions:

WR1 The unit shall be a volume_unit.

20.4.93 volume_unit

A **volume_unit** is a type of derived_unit used to measure the extent of a solid.

EXPRESS specification:

```
ENTITY volume_unit
  SUBTYPE OF (derived_unit);
WHERE
  WR1: derive_dimensional_exponents(SELF) = dimensional_exponents(3.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0);
END_ENTITY;
```

Attribute definitions:

WR1 The dimensional exponent of length shall be equal to three and all the other dimensional exponents shall be equal to zero.

 © ISO 2019 – All rights reserved

20.5 measure_schema function definitions

20.5.1 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
    REPEAT i := LOINDEX(x\derived_unit.elements) TO HIINDEX(x\derived_unit.elements);
      result.length_exponent := result.length_exponent + x\derived_unit.
elements[i]\derived_unit_element.exponent * x\derived_unit.elements[i]\derived_unit_
element.unit\named_unit.dimensions.length_exponent;
      result.mass_exponent := result.mass_exponent + x\derived_unit.elements[i]\derived_
unit_element.exponent * x\derived_unit.elements[i]\derived_unit_element.unit\named_unit.
dimensions.mass_exponent;
      result.time_exponent := result.time_exponent + x\derived_unit.elements[i]\derived_
unit_element.exponent * x\derived_unit.elements[i]\derived_unit_element.unit\named_unit.
dimensions.time_exponent;
      result.electric_current_exponent := result.electric_current_exponent + x\derived_
unit.elements[i]\derived_unit_element.exponent * x\derived_unit.elements[i]\derived_unit_
element.unit\named_unit.dimensions.electric_current_exponent;
      result.thermodynamic_temperature_exponent := result.thermodynamic_temperature_
exponent + x\derived_unit.elements[i]\derived_unit_element.exponent * x\derived_unit.
elements[i]\derived_unit_element.unit\named_unit.dimensions.thermodynamic_temperature_
exponent;
      result.amount_of_substance_exponent := result.amount_of_substance_exponent
+ x\derived_unit.elements[i]\derived_unit_element.exponent * x\derived_unit.
elements[i]\derived_unit_element.unit\named_unit.dimensions.amount_of_substance_exponent;
      result.luminous_intensity_exponent := result.luminous_intensity_exponent
+ x\derived_unit.elements[i]\derived_unit_element.exponent * x\derived_unit.
elements[i]\derived_unit_element.unit\named_unit.dimensions.luminous_intensity_exponent;
    END_REPEAT;
  ELSE
    result := x\named_unit.dimensions;
  END_IF;
  RETURN (result);
END_FUNCTION;
```

20.5.1.1 Parameters

20.5.1.1.1 x

(input) the unit that the dimensional_exponents are being derived from.

EXPRESS specification:

```
x : unit
```

20.5.1.2 Variables

20.5.1.2.1 result

EXPRESS specification:

```
result : dimensional_exponents := dimensional_exponents(0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0);
```

© ISO 2019 – All rights reserved

20.5.2 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, 2.0, 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 (?);
```


© ISO 2019 – All rights reserved

```
  END_CASE;
END_FUNCTION;
```

20.5.2.1 Parameters

20.5.2.1.1 n

(input) the name of the si_unit_name for which the dimensional_exponents will be returned.

EXPRESS specification:

```
n : si_unit_name
```

20.5.3 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;
  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
```

© ISO 2019 – All rights reserved

```
      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;
  IF 'MEASURE_SCHEMA.ACCELERATION_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(1.0, 0.0, -
2.0, 0.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.CAPACITANCE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(-2.0, -
1.0, 4.0, 2.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.ELECTRIC_CHARGE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(0.0, 0.0,
1.0, 1.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.CONDUCTANCE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(-2.0, -
1.0, 3.0, 2.0, 0.0, 0.0, 0.0) THEN
```

 © ISO 2019 – All rights reserved

```
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.ELECTRIC_POTENTIAL_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 1.0, -
3.0, -1.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.ENERGY_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 1.0, -
2.0, 0.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.FORCE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(1.0, 1.0, -
2.0, 0.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.FREQUENCY_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.ILLUMINANCE_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, 1.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.INDUCTANCE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 1.0, -
2.0, -2.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.LUMINOUS_FLUX_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.MAGNETIC_FLUX_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 1.0, -
2.0, -1.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.MAGNETIC_FLUX_DENSITY_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(0.0, 1.0, -
2.0, -1.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.POWER_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 1.0, -
3.0, 0.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.PRESSURE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(-1.0, 1.0,
-2.0, 0.0, 0.0, 0.0, 0.0) THEN
```

© ISO 2019 – All rights reserved

```
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.RESISTANCE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 1.0, -
3.0, -2.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.VELOCITY_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(1.0, 0.0, -
1.0, 0.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.RADIOACTIVITY_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.ABSORBED_DOSE_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 0.0, -
2.0, 0.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'MEASURE_SCHEMA.DOSE_EQUIVALENT_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(2.0, 0.0, -
2.0, 0.0, 0.0, 0.0, 0.0) THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  RETURN (TRUE);
END_FUNCTION;
```

20.5.3.1 Parameters

20.5.3.1.1 m

(input) the candidate measure_with_unit that is to be checked.

EXPRESS specification:

```
m : measure_with_unit
```

21 person_organization_schema

21.1 General

The subject of the **person_organization_schema** is the description of information that identifies people and organizations.

EXPRESS specification:

```
SCHEMA person_organization_schema '{iso standard 10303 part(41) version(8) object(1)
person_organization_schema(17)}';

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   description_attribute_select,
   generic_attribute,
   generic_attribute_assignment,
   generic_attribute_list_assignment,
   generic_attribute_list_select,
```

 © ISO 2019 – All rights reserved

```
   generic_attribute_select,
   get_description_value,
   get_id_value,
   get_name_value,
   get_role,
   id_attribute,
   id_attribute_select,
   name_attribute,
   name_attribute_select,
   object_role,
   role_association);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

21.2 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.

21.3 person_organization_schema type definitions

21.3.1 person_organization_select

The **person_organization_select** type allows for the designation of an organization, a person or a person_and_organization.

EXPRESS specification:

```
TYPE person_organization_select = SELECT
  (person,
   organization,
   person_and_organization);
END_TYPE;
```

21.3.2 pos_description_attribute_select

EXPRESS specification:

```
TYPE pos_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (person_and_organization_role,
   person_and_organization,
   person_role,
   organization_role,
   organizational_project);
END_TYPE;
```

21.3.3 pos_generic_attribute_select

EXPRESS specification:

```
TYPE pos_generic_attribute_select = SELECT BASED_ON generic_attribute_select WITH
  (address);
END_TYPE;
```

21.3.4 pos_generic_attribute_list_select

EXPRESS specification:

© ISO 2019 – All rights reserved

```
TYPE pos_generic_attribute_list_select = SELECT BASED_ON generic_attribute_list_select
WITH
  (address,
   person);
END_TYPE;
```

21.3.5 pos_id_attribute_select

EXPRESS specification:

```
TYPE pos_id_attribute_select = SELECT BASED_ON id_attribute_select WITH
  (address,
   organizational_project);
END_TYPE;
```

21.3.6 pos_name_attribute_select

EXPRESS specification:

```
TYPE pos_name_attribute_select = SELECT BASED_ON name_attribute_select WITH
  (address,
   person_and_organization);
END_TYPE;
```

21.4 person_organization_schema entity definitions

21.4.1 address

An **address** is the information necessary for communicating, using one or more communication methods.

NOTE 1 In addition to the attributes defined in **address** additional generic attributes may be assigned using generic_attribute_assignment.

EXAMPLE 1 A generic attribute named “LinkedIn” with a value of “Fred.Bloggs” could be used to add a LinkedIn account to address information.

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;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

internal_location an organization-defined address for internal mail delivery. The value of this attribute need not be specified.

street_number the number of a location on a street. The value of this attribute need not be specified.

street the name of a street. The value of this attribute need not be specified.

postal_box the number of a postal box. The value of this attribute need not be specified.

town the name of a town. The value of this attribute need not be specified.

region the name of a region. The value of this attribute need not be specified.

EXAMPLE 2 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. The value of this attribute need not be specified.

country the name of a country. The value of this attribute need not be specified.

facsimile_number the number at which facsimiles may be received. The value of this attribute need not be specified.

telephone_number the number at which telephone calls may be received. The value of this attribute need not be specified.

electronic_mail_address the electronic address at which electronic mail may be received. The value of this attribute need not be specified.

telex_number the number at which telex messages may be received. The value of this attribute need not be specified.

name the label that specifies the name of the location identified by the **address**.

EXAMPLE 3 "White House", "Buckingham Palace", "Kremlin".

url the text that specifies the uniform resource locator associated with the **address**.

NOTE 2 The format of uniform resource locators is defined in IETF RFC 1738 [4].

EXAMPLE 4 URLs are used to identify items such as world wide web sites and file transfer protocol (FTP) servers.

Attribute definitions:

WR1 At least one of the attributes shall have a value.

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

21.4.2 organization

An **organization** is an administrative structure.

EXPRESS specification:

```
ENTITY organization;
  id : OPTIONAL identifier;
```

© ISO 2019 – All rights reserved

```
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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 this attribute need not be specified.

21.4.3 organization_relationship

An **organization_relationship** is a relationship between 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;
```

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.

© ISO 2019 – All rights reserved

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.

21.4.4 organization_role

An **organization_role** is a purpose for an **organization** and provides identification and description of the purpose.

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;
```

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.

Attribute definitions:

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 basic_attribute_schema 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.

21.4.5 organization_type

An **organization_type** is a recognized kind of organization.

EXAMPLE A national standards body is a type of organization.

© ISO 2019 – All rights reserved

NOTE A relationship between an **organization_type** object and one or more organization objects is established by the declaration of a subtype of organization_type_assignment that references an organization via its items attribute.

EXPRESS specification:

```
ENTITY organization_type;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **organization_type**.

name the label by which the **organization_type** is known.

description the text that characterizes the **organization_type**. The value of this attribute need not be specified.

21.4.6 organization_type_relationship

An **organization_type_relationship** is a relationship between two instances of the organization_type entity data type and provides for the identification, naming, and description of that relationship.

EXPRESS specification:

```
ENTITY organization_type_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_organization_type : organization_type;
  related_organization_type : organization_type;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **organization_type_relationship**.

name the label by which the **organization_type_relationship** is known.

description the text that characterizes the **organization_type_relationship**. The value of this attribute need not be specified.

relating_organization_type one of the instances of organization_type that is a part of the relationship.

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.

related_organization_type the other instance of organization_type 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 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.

 © ISO 2019 – All rights reserved

21.4.7 organizational_address

An **organizational_address** is a type of address for one or more organizations.

EXPRESS specification:

```
ENTITY organizational_address
  SUBTYPE OF (address);
  organizations : SET [1:?] OF organization;
  description : OPTIONAL text;
END_ENTITY;
```

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.

21.4.8 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;
```

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.

Attribute definitions:

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

© ISO 2019 – All rights reserved

NOTE 3 The id_attribute data type is defined in basic_attribute_schema 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.

21.4.9 organizational_project_relationship

An **organizational_project_relationship** is a relationship between 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;
```

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.

21.4.10 person

A **person** is an individual human being.

 © ISO 2019 – All rights reserved

NOTE 1 In addition to the attributes defined in **person** additional lists of attributes may be assigned using generic_attribute_list_assignment.

EXAMPLE 1 For example a generic attribute list named “Previous name” with a value of [“Honourific”:”Mr”, “Given name”:”Fred”, “Family Name”:”Blogs”] may be assigned.

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;
```

Attribute definitions:

id the identifier that distinguishes the **person**.

NOTE 2 The identification of a person is usually context dependent.

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 2 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 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.

last_name the **person**‘s surname. The value of this attribute need not be specified.

first_name the first element of the **person**‘s list of forenames. The value of this attribute need not be specified.

middle_names the **person**‘s other forenames, if there are any. The value of this attribute need not be specified.

prefix_titles the text that specifies the **person**‘s social or professional standing and appear before his or her names. The value of this attribute need not be specified.

suffix_titles the text that specifies the **person**‘s social or professional standing and appear after his or her names. The value of this attribute need not be specified.

Attribute definitions:

WR1 Either the last name, first name or the last and first name shall be defined.

21.4.11 person_and_organization

A **person_and_organization** is a person in an organization.

EXPRESS specification:

© ISO 2019 – All rights reserved

```
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;
```

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.

Attribute definitions:

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 basic_attribute_schema 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 basic_attribute_schema 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.

21.4.12 person_and_organization_role

A **person_and_organization_role** is a purpose for a **person_and_organization** and provides identification and description of the purpose.

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
```

 © ISO 2019 – All rights reserved

```
  description : text := get_description_value(SELF);
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'BASIC_ATTRIBUTE_SCHEMA.' + 'DESCRIPTION_ATTRIBUTE.DESCRIBED_
ITEM')) <= 1;
END_ENTITY;
```

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.

Attribute definitions:

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 basic_attribute_schema 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.

21.4.13 person_role

A **person_role** is a purpose for a person and provides identification and description of the purpose.

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;
```

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 label that characterizes the **person_role**.

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

© ISO 2019 – All rights reserved

Attribute definitions:

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 basic_attribute_schema 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.

21.4.14 person_type

A **person_type** is a category of people having common characteristics. // <note>

EXPRESS specification:

```
ENTITY person_type;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type**.

name the label by which the **person_type** is known.

description the text that characterizes the **person_type**. The value of this attribute need not be specified.

21.4.15 person_type_definition

A **person_type_definition** is a characterization of type of person.

EXPRESS specification:

```
ENTITY person_type_definition;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  formation : person_type_definition_formation;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type_definition**.

name the label by which the **person_type_definition** is known.

description the text that characterizes the **person_type_definition**. The value of this attribute need not be specified.

formation the person_type_definition_formation to which the **person_type_definition** relates.

21.4.16 person_type_definition_formation

A **person_type_definition_formation** is a collector of definitions of a type of person.

 © ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY person_type_definition_formation;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  of_person_type : person_type;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type_definition_formation**.

name the label by which the **person_type_definition_formation** is known.

description the text that characterizes the **person_type_definition_formation**. The value of this attribute need not be specified.

of_person_type the person_type to which the **person_type_definition_formation** belongs.

21.4.17 person_type_definition_relationship

A **person_type_definition_relationship** is a relationship between two instances of the person_type_definition entity data type and provides for the identification, naming, and description of that relationship.

EXPRESS specification:

```
ENTITY person_type_definition_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_person_type_definition : person_type_definition;
  related_person_type_definition : person_type_definition;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **person_type_definition_relationship**.

name the label by which the **person_type_definition_relationship** is known.

description the text that characterizes the **person_type_definition_relationship**. The value of this attribute need not be specified.

relating_person_type_definition one of the instances of person_type_definition that is a part of the relationship.

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.

related_person_type_definition the other instance of person_type_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.

© ISO 2019 – All rights reserved

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.

21.4.18 personal_address

A **personal_address** is a type of address, which 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;
```

Attribute definitions:

people the people who reside at the address.

description the text that characterizes the **personal_address**. The value of this attribute need not be specified.

21.4.19 position_in_organization

A **position_in_organization** is a specific function or job performed by a person in a particular organization. It is defined by responsibilities and activities. A position that is not fulfilled by a person is a vacancy.

EXAMPLE The chairperson of SC4.

EXPRESS specification:

```
ENTITY position_in_organization;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **position_in_organization**.

name the label by which the **position_in_organization** is known.

description the text that characterizes the **position_in_organization**. The value of this attribute need not be specified.

21.4.20 position_in_organization_relationship

A **position_in_organization_relationship** is a relationship between two instances of the position_in_organization entity data type and provides for the identification, naming, and description of that relationship.

EXPRESS specification:

```
ENTITY position_in_organization_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_position_in_organization : position_in_organization;
```

 © ISO 2019 – All rights reserved

```
  related_position_in_organization : position_in_organization;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **position_in_organization_relationship**.

name the label by which the **position_in_organization_relationship** is known.

description the text that characterizes the **position_in_organization_relationship**. The value of this attribute need not be specified.

relating_position_in_organization one of the instances of position_in_organization that is a part of the relationship.

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.

related_position_in_organization the other instance of position_in_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 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.

21.4.21 position_in_organization_type

A **position_in_organization_type** is a generic function or job that is performed in organizations.

EXAMPLE Technical Director is a position in a company.

NOTE A relationship between a **position_in_organization_type** object and one or more position_in_organization objects is established by the declaration of a subtype of position_in_organization_type_assignment that references a position_in_organization via its items attribute.

EXPRESS specification:

```
ENTITY position_in_organization_type;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **position_in_organization_type**.

name the label by which the **position_in_organization_type** is known.

description the text that characterizes the **position_in_organization_type**. The value of this attribute need not be specified.

© ISO 2019 – All rights reserved

21.5 person_organization_schema function definitions

21.5.1 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 an 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.relating_organization IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(org <* bag_to_set(USEDIN(relation.relating_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.relating_
organization, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

21.5.1.1 Parameters

21.5.1.1.1 relation

(input) the candidate organization_relationship to be checked.

EXPRESS specification:

```
relation : organization_relationship
```

21.5.1.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF organization
```

21.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the organization_relationship entity.

EXPRESS specification:

 © ISO 2019 – All rights reserved

```
specific_relation : STRING
```

21.5.1.2 Variables

21.5.1.2.1 x

EXPRESS specification:

```
x : SET OF organization_relationship;
```

21.5.1.3 Statements

21.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_organization_relationship(x[i], relatives + relation.relating_
organization, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

21.5.2 acyclic_organization_type_relationship

The **acyclic_organization_type_relationship** function determines whether the graph of instances of the entity data type organization_type that contains relation as one of its links contains a cycle. This function may be used to evaluate either an organization_type_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_type_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_organization_type_relationship(relation : organization_type_relationship;
                                                relatives : SET OF organization_type;
                                                specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF organization_type_relationship;
  END_LOCAL;
  IF relation.relating_organization_type IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(orgtyp <* bag_to_set(USEDIN(relation.relating_organization_type, ('PERSON_
ORGANIZATION_SCHEMA.' + 'ORGANIZATION_TYPE_RELATIONSHIP.') + 'RELATED_ORGANIZATION_
TYPE')) | specific_relation IN TYPEOF(orgtyp));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_organization_type_relationship(x[i], relatives + relation.relating_
organization_type, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

21.5.2.1 Parameters

21.5.2.1.1 relation

(input) the candidate organization_type_relationship to be checked.

© ISO 2019 – All rights reserved

EXPRESS specification:

```
relation : organization_type_relationship
```

21.5.2.1.2 relatives

(input) the set of instances of the entity data type organization_type that the function is searching for in the relating_organization_type parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET OF organization_type
```

21.5.2.1.3 specific_relation

(input) the fully qualified name of a subtype of the organization_type_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

21.5.2.2 Variables

21.5.2.2.1 x

EXPRESS specification:

```
x : SET OF organization_type_relationship;
```

21.5.2.3 Statements

21.5.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_organization_type_relationship(x[i], relatives + relation.relating_
organization_type, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

21.5.3 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 an 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;
```

 © ISO 2019 – All rights reserved

```
  END_LOCAL;
  IF relation.relating_organizational_project IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(op <* bag_to_set(USEDIN(relation.relating_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.
relating_organizational_project, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

21.5.3.1 Parameters

21.5.3.1.1 relation

(input) the candidate organizational_project_relationship to be checked.

EXPRESS specification:

```
relation : organizational_project_relationship
```

21.5.3.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF organizational_project
```

21.5.3.1.3 specific_relation

(input) the fully qualified entity name of a type of organizational_project_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

21.5.3.2 Variables

21.5.3.2.1 x

EXPRESS specification:

```
x : SET OF organizational_project_relationship;
```

21.5.3.3 Statements

21.5.3.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_organizational_project_relationship(x[i], relatives + relation.relating_
organizational_project, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

© ISO 2019 – All rights reserved

21.5.4 acyclic_person_type_definition_relationship

The **acyclic_person_type_definition_relationship** function determines whether the graph of instances of the entity data type person_type_definition that contains relation as one of its links contains a cycle. This function may be used to evaluate either a person_type_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 person_type_definition_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_person_type_definition_relationship(relation : person_type_definition_
relationship;
                                                     relatives : SET OF person_type_
definition;
                                                     specific_relation : STRING) :
BOOLEAN;
  LOCAL
    x : SET OF person_type_definition_relationship;
  END_LOCAL;
  IF relation.relating_person_type_definition IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(ptdef <* bag_to_set(USEDIN(relation.relating_person_type_definition,
('PERSON_ORGANIZATION_SCHEMA.' + 'PERSON_TYPE_DEFINITION_RELATIONSHIP.') + 'RELATED_
PERSON_TYPE_DEFINITION')) | specific_relation IN TYPEOF(ptdef));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_person_type_definition_relationship(x[i], relatives + relation.
relating_person_type_definition, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

21.5.4.1 Parameters

21.5.4.1.1 relation

(input) the candidate person_type_definition_relationship to be checked.

EXPRESS specification:

```
relation : person_type_definition_relationship
```

21.5.4.1.2 relatives

(input) the set of instances of the entity data type person_type_definition that the function is searching for in the relating_person_type_definition parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET OF person_type_definition
```

21.5.4.1.3 specific_relation

(input) the fully qualified name of a subtype of the person_type_definition_relationship entity.

EXPRESS specification:

 © ISO 2019 – All rights reserved

```
specific_relation : STRING
```

21.5.4.2 Variables

21.5.4.2.1 x

EXPRESS specification:

```
x : SET OF person_type_definition_relationship;
```

21.5.4.3 Statements

21.5.4.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_person_type_definition_relationship(x[i], relatives + relation.relating_
person_type_definition, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

21.5.5 acyclic_position_in_organization_relationship

The **acyclic_position_in_organization_relationship** function determines whether the graph of instances of the entity data type position_in_organization that contains relation as one of its links contains a cycle. This function may be used to evaluate either a position_in_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 position_in_organization_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_position_in_organization_relationship(relation : position_in_
organization_relationship;
                                                       relatives : SET OF position_in_
organization;
                                                       specific_relation : STRING) :
BOOLEAN;
  LOCAL
    x : SET OF position_in_organization_relationship;
  END_LOCAL;
  IF relation.relating_position_in_organization IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(piorg <* bag_to_set(USEDIN(relation.relating_position_in_organization,
('PERSON_ORGANIZATION_SCHEMA.' + 'POSITION_IN_ORGANIZATION_RELATIONSHIP.') + 'RELATED_
POSITION_IN_ORGANIZATION')) | specific_relation IN TYPEOF(piorg));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_position_in_organization_relationship(x[i], relatives + relation.
relating_position_in_organization, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

© ISO 2019 – All rights reserved

21.5.5.1 Parameters

21.5.5.1.1 relation

(input) the candidate position_in_organization_relationship to be checked.

EXPRESS specification:

```
relation : position_in_organization_relationship
```

21.5.5.1.2 relatives

(input) the set of instances of the entity data type position_in_organization that the function is searching for in the relating_position_in_organization parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET OF position_in_organization
```

21.5.5.1.3 specific_relation

(input) the fully qualified name of a subtype of the position_in_organization_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

21.5.5.2 Variables

21.5.5.2.1 x

EXPRESS specification:

```
x : SET OF position_in_organization_relationship;
```

21.5.5.3 Statements

21.5.5.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_position_in_organization_relationship(x[i], relatives + relation.
relating_position_in_organization, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

22 product_definition_schema

22.1 General

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.

EXPRESS specification:

```
SCHEMA product_definition_schema '{iso standard 10303 part(41) version(9) object(1)
product_definition_schema(18)}';
```

 © ISO 2019 – All rights reserved

```
REFERENCE FROM application_context_schema
  (product_context,
   product_definition_context);

REFERENCE FROM basic_attribute_schema
  (get_id_value,
   get_name_value,
   id_attribute,
   id_attribute_select,
   name_attribute,
   name_attribute_select);

REFERENCE FROM document_schema
  (document);

REFERENCE FROM effectivity_schema
  (effectivity);

REFERENCE FROM external_reference_schema
  (external_source);

REFERENCE FROM product_property_definition_schema
  (characterized_object);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

22.2 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.

© ISO 2019 – All rights reserved

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 an 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.

22.3 product_definition_schema type definitions

22.3.1 product_definition_or_reference

EXPRESS specification:

```
TYPE product_definition_or_reference = EXTENSIBLE GENERIC_ENTITY SELECT
  (product_definition,
   generic_product_definition_reference);
END_TYPE;
```

22.3.2 pds_id_attribute_select

EXPRESS specification:

```
TYPE pds_id_attribute_select = SELECT BASED_ON id_attribute_select WITH
  (product_category);
END_TYPE;
```

22.3.3 pds_name_attribute_select

EXPRESS specification:

```
TYPE pds_name_attribute_select = SELECT BASED_ON name_attribute_select WITH
  (product_definition,
   product_definition_substitute);
END_TYPE;
```

 © ISO 2019 – All rights reserved

22.3.4 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;
```

22.4 product_definition_schema entity definitions

22.4.1 alternative_solution_relationship

An **alternative_solution_relationship** is a product_definition_formation_relationship that specifies the type of the relationship.

EXPRESS specification:

```
ENTITY alternative_solution_relationship
  SUBTYPE OF (product_definition_formation_relationship);
  relation_type : STRING;
WHERE
  WR1: acyclic_alternative_solution_relationship(SELF, [SELF\product_definition_
formation_relationship.related_product_definition_formation], 'PRODUCT_DEFINITION_SCHEMA.
ALTERNATIVE_SOLUTION_RELATIONSHIP');
END_ENTITY;
```

Attribute definitions:

relation_type the meaning of the relationship.

Attribute definitions:

WR1 An **alternative_solution_relationship** shall not participate in its own definition.

22.4.2 characterized_product

A **characterized_product** is a type of characterized_object and a type of product for the purpose of characterizing a product.

NOTE Properties that are assigned using **characterized_object** are outside version control, whereas properties added via **product_definition** are under version control.

EXPRESS specification:

```
ENTITY characterized_product
  SUBTYPE OF (characterized_object, product);
END_ENTITY;
```

22.4.3 generic_product_definition_reference

A **generic_product_definition_reference** is a reference to an externally defined product_definition that may or may not have a local representation. A **generic_product_definition_reference** is either a product_definition_reference or a product_definition_reference_with_local_representation.

EXPRESS specification:

```
ENTITY generic_product_definition_reference
```

© ISO 2019 – All rights reserved

```
  ABSTRACT SUPERTYPE OF (ONEOF(product_definition_reference, product_definition_reference_
with_local_representation));
  source : external_source;
END_ENTITY;
```

Attribute definitions:

source defines the location which contains the original definition of the product_definition.

22.4.4 product

A **product** is a representation of a product or a type of product

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;
```

Attribute definitions:

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

 © ISO 2019 – All rights reserved

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.

22.4.5 product_category

A **product_category** is a classification that applies to products.

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;
```

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**.

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.

© ISO 2019 – All rights reserved

Attribute definitions:

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 basic_attribute_schema 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.

22.4.6 product_category_relationship

A **product_category_relationship** is a hierarchical relationship of one category with another and provides a description of their relationship. If a product belongs to a product_category, it also belongs to all of the parent categories.

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_categorys 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_relationships**, one **product_category_relationship** between 'piping part' and 'cold water pipe' and the other **product_category_relationship** between 'piping part' and 'hot water pipe'.

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;
```

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**.

 © ISO 2019 – All rights reserved

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.

Attribute definitions:

WR1 A graph of product_category instances shall not be cyclic.

22.4.7 product_definition

A **product_definition** is a representation of 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.

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

EXAMPLE 3 The designed shape of an aeroplane can be identified by an instance of the product_definition_shape entity data type 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;
```

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.

© ISO 2019 – All rights reserved

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.

Attribute definitions:

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

NOTE 6 The name_attribute data type is defined in basic_attribute_schema 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.

22.4.8 product_definition_context_association

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

NOTE 1 A **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 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;
```

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.

 © ISO 2019 – All rights reserved

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.

22.4.9 product_definition_context_role

A **product_definition_context_role** is a specification of the purpose of a product_definition_context.

EXPRESS specification:

```
ENTITY product_definition_context_role;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

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.

22.4.10 product_definition_effectivity

A **product_definition_effectivity** is a type of 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 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;
```

Attribute definitions:

usage the product_definition_relationship that defines the context of the effectivity.

Attribute definitions:

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

22.4.11 product_definition_formation

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

© ISO 2019 – All rights reserved

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 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;
```

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.

22.4.12 final_solution

A **final_solution** is a type of product_definition_formation and is a set of additional sensual characteristics that can be applied to an occurrence that represents a neutral part in order to complete its definition.

EXPRESS specification:

```
ENTITY final_solution
  SUBTYPE OF (product_definition_formation);
  status : STRING;
INVERSE
  specification : SET [1:?] OF product_definition FOR formation;
END_ENTITY;
```

Attribute definitions:

status the text string that describes the level of completion between the neutral part and the final part.

© ISO 2019 – All rights reserved

specification the set of instances of the entity data type product_definition that defines the various aspects of the **final_solution**.

22.4.13 product_definition_formation_relationship

A **product_definition_formation_relationship** is a relationship between 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;
```

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.

© ISO 2019 – All rights reserved

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.

22.4.14 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;
```

Attribute definitions:

make_or_buy the characterization of the source of the product_definition_formation.

22.4.15 product_definition_relationship

A **product_definition_relationship** is a relationship between two instances of the entity data type product_definition or generic_product_definition_reference 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;
```

 © ISO 2019 – All rights reserved

```
  name : label;
  description : OPTIONAL text;
  relating_product_definition : product_definition_or_reference;
  related_product_definition : product_definition_or_reference;
END_ENTITY;
```

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.

NOTE 9 In the context of an ISO 10303 application protocol, 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 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 In the context of an ISO 10303 application protocol, the **related_product_definition** may be the product_definition that is an element of the assembly.

22.4.16 product_definition_relationship_relationship

A **product_definition_relationship_relationship** is a relationship between two instances of the entity data type product_definition_relationship and provides an identification and description of this relationship.

EXPRESS specification:

```
ENTITY product_definition_relationship_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating : product_definition_relationship;
  related : product_definition_relationship;
```

© ISO 2019 – All rights reserved

```
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **product_definition_relationship_relationship**.

name the label by which the **product_definition_relationship_relationship** is known.

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

relating one of the instances of product_definition that is a part of the relationship.

related 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.

22.4.17 product_definition_reference

A **product_definition_reference** is a type of generic_product_definition_reference that identifies the externally defined product_definition by providing the key attributes necessary to unambiguously identify it and the associated product_definition_formation, product and organization that owns the product id.

EXPRESS specification:

```
ENTITY product_definition_reference
  SUBTYPE OF (generic_product_definition_reference);
  product_id : identifier;
  product_definition_formation_id : identifier;
  product_definition_id : identifier;
  id_owning_organization_name : OPTIONAL label;
END_ENTITY;
```

Attribute definitions:

product_id specifies the identifier of the externally defined product, where the referenced product_definition is assigned to one of its product_definition_formation.

product_definition_formation_id specifies the identifier of the externally defined product_definition_formation.

product_definition_id specifies the identifier of the externally defined product_definition.

id_owning_organization_name specifies the id owner of the externally defined product, realised by an applied organization_assignment with the role 'id owner'.

22.4.18 product_definition_reference_with_local_representation

A **product_definition_reference_with_local_representation** is a type of generic_product_definition_reference that represents an externally defined product_definition by a local product_definition.

EXPRESS specification:

```
ENTITY product_definition_reference_with_local_representation
  SUBTYPE OF (generic_product_definition_reference, product_definition);
END_ENTITY;
```


© ISO 2019 – All rights reserved

22.4.19 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 is 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;
```

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.

Attribute definitions:

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 basic_attribute_schema of this part of ISO 10303

© ISO 2019 – All rights reserved

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

22.4.20 product_definition_with_associated_documents

A **product_definition_with_associated_documents** is a type of 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;
```

Attribute definitions:

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

22.4.21 product_related_product_category

A **product_related_product_category** is a type of product_category that identifies 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;
```

Attribute definitions:

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

22.4.22 product_relationship

A **product_relationship** is a relationship between 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;
```


© ISO 2019 – All rights reserved

```
END_ENTITY;
```

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.

22.5 product_definition_schema function definitions

22.5.1 acyclic_alternative_solution_relationship

The **acyclic_alternative_solution_relationship** function determines whether the graph of instance 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 alternative_solution_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_alternative_solution_relationship(relation : alternative_solution_
relationship;
                                                   relatives : SET [1:?] OF product_
definition_formation;
                                                   specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF alternative_solution_relationship;
  END_LOCAL;
  IF relation.relating_product_definition_formation IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(asr <* bag_to_set(USEDIN(relation.relating_product_definition_formation,
('PRODUCT_DEFINITION_SCHEMA.' + 'ALTERNATIVE_SOLUTION_RELATIONSHIP.') + 'RELATED_PRODUCT_
DEFINITION_FORMATION')) | specific_relation IN TYPEOF(asr));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_alternative_solution_relationship(x[i], relatives + relation.relating_
product_definition_formation, specific_relation) THEN
      RETURN (FALSE);
```

© ISO 2019 – All rights reserved

```
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

22.5.1.1 Parameters

22.5.1.1.1 relation

(input) the candidate alternative_solution_relationship to be checked.

EXPRESS specification:

```
relation : alternative_solution_relationship
```

22.5.1.1.2 relatives

(input) the set of instances of the entity data type product_definition_formation for which the function is searching in the relating_product_definition_formation parameter of the relation argument.

EXPRESS specification:

```
relatives : SET [1:?] OF product_definition_formation
```

22.5.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the alternative_solution_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

22.5.1.2 Variables

22.5.1.2.1 x

EXPRESS specification:

```
x : SET OF alternative_solution_relationship;
```

22.5.1.3 Statements

22.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_alternative_solution_relationship(x[i], relatives + relation.relating_
product_definition_formation, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

22.5.2 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;
```

 © ISO 2019 – All rights reserved

```
                                                    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;
```

22.5.2.1 Parameters

22.5.2.1.1 relation

(input) the candidate product_category_relationship to be checked.

EXPRESS specification:

```
relation : product_category_relationship
```

22.5.2.1.2 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.

EXPRESS specification:

```
children : SET OF product_category
```

22.5.2.2 Variables

22.5.2.2.1 x

EXPRESS specification:

```
x : SET OF product_category_relationship;
```

22.5.2.2.2 local_children

EXPRESS specification:

```
local_children : SET OF product_category;
```

22.5.2.3 Statements

22.5.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(children);
  IF relation.category :=: children[i] THEN
    RETURN (FALSE);
```

© ISO 2019 – All rights reserved

```
  END_IF;
END_REPEAT;
```

22.5.3 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.relating_product_definition_formation IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(pdf <* bag_to_set(USEDIN(relation.relating_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.
relating_product_definition_formation, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

22.5.3.1 Parameters

22.5.3.1.1 relation

(input) the candidate product_definition_formation_relationship to be checked.

EXPRESS specification:

```
relation : product_definition_formation_relationship
```

22.5.3.1.2 relatives

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

EXPRESS specification:

```
relatives : SET [1:?] OF product_definition_formation
```


© ISO 2019 – All rights reserved

22.5.3.1.3 specific_relation

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

EXPRESS specification:

```
specific_relation : STRING
```

22.5.3.2 Variables

22.5.3.2.1 x

EXPRESS specification:

```
x : SET OF product_definition_formation_relationship;
```

22.5.3.3 Statements

22.5.3.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_product_definition_formation_relationship(x[i], relatives + relation.
relating_product_definition_formation, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

22.5.4 acyclic_product_definition_relationship

The **acyclic_product_definition_relationship** function determines whether the graph of instances of the entity data type product_definition 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_or_reference;
                                                 specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF product_definition_relationship;
  END_LOCAL;
  IF relation.relating_product_definition IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(pd <* bag_to_set(USEDIN(relation.relating_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.relating_
product_definition, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
```

© ISO 2019 – All rights reserved

```
END_FUNCTION;
```

22.5.4.1 Parameters

22.5.4.1.1 relation

(input) the candidate product_definition_relationship to be checked.

EXPRESS specification:

```
relation : product_definition_relationship
```

22.5.4.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF product_definition_or_reference
```

22.5.4.1.3 specific_relation

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

EXPRESS specification:

```
specific_relation : STRING
```

22.5.4.2 Variables

22.5.4.2.1 x

EXPRESS specification:

```
x : SET OF product_definition_relationship;
```

22.5.4.3 Statements

22.5.4.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_product_definition_relationship(x[i], relatives + relation.relating_
product_definition, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

22.5.5 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;
```

 © ISO 2019 – All rights reserved

```
                                        relatives : SET [1:?] OF product;
                                        specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF product_relationship;
  END_LOCAL;
  IF relation.relating_product IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(prod <* bag_to_set(USEDIN(relation.relating_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.relating_product,
specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

22.5.5.1 Parameters

22.5.5.1.1 relation

(input) the candidate product_relationship to be checked.

EXPRESS specification:

```
relation : product_relationship
```

22.5.5.1.2 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.

EXPRESS specification:

```
relatives : SET [1:?] OF product
```

22.5.5.1.3 specific_relation

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

EXPRESS specification:

```
specific_relation : STRING
```

22.5.5.2 Variables

22.5.5.2.1 x

EXPRESS specification:

```
x : SET OF product_relationship;
```

22.5.5.3 Statements

22.5.5.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_product_relationship(x[i], relatives + relation.relating_product,
specific_relation) THEN
    RETURN (FALSE);
  END_IF;
```

© ISO 2019 – All rights reserved

```
END_REPEAT;
```

22.5.6 categories_of_product

The **categories_of_product** function returns the set of product category names associated with the argument. The function evaluates the bag of instances of product_category that are associated with the input product through an instance of product_related_product_category. The function returns the SET of STRING values stored in the name attribute of the instances of product_category in the bag.

EXPRESS specification:

```
FUNCTION categories_of_product(obj : product) : SET OF STRING;
  LOCAL
    category_assignments : BAG OF product_category;
    categories : SET OF STRING := [];
  END_LOCAL;
  category_assignments := USEDIN(obj, 'PRODUCT_DEFINITION_SCHEMA.' + 'PRODUCT_RELATED_
PRODUCT_CATEGORY.PRODUCTS');
  REPEAT i := LOINDEX(category_assignments) TO HIINDEX(category_assignments) BY 1;
    categories := categories + category_assignments[i].name;
  END_REPEAT;
  RETURN (categories);
END_FUNCTION;
```

22.5.6.1 Parameters

22.5.6.1.1 obj

the product whose categories are searched.

EXPRESS specification:

```
obj : product
```

22.5.6.2 Variables

22.5.6.2.1 category_assignments

EXPRESS specification:

```
category_assignments : BAG OF product_category;
```

22.5.6.2.2 categories

EXPRESS specification:

```
categories : SET OF STRING := [];
```

22.5.6.3 Statements

22.5.6.3.1 i

EXPRESS specification:

```
REPEAT i := LOINDEX(category_assignments) TO HIINDEX(category_assignments) BY 1;
  categories := categories + category_assignments[i].name;
END_REPEAT;
```

22.5.7 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.

© ISO 2019 – All rights reserved

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;
  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;
```

22.5.7.1 Parameters

22.5.7.1.1 c_def_instance

the candidate product to be checked.

EXPRESS specification:

```
c_def_instance : product
```

22.5.7.2 Variables

22.5.7.2.1 pd_set

EXPRESS specification:

```
pd_set : SET OF product_definition_formation := [];
```

22.5.7.2.2 pdr_set

EXPRESS specification:

```
pdr_set : SET OF product_definition := [];
```

22.5.7.3 Statements

22.5.7.3.1 i

EXPRESS specification:

```
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;
```

23 product_property_definition_schema

23.1 General

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

EXPRESS specification:

```
SCHEMA product_property_definition_schema '{iso standard 10303 part(41) version(9)
object(1) product_property_definition_schema(19)}';
```

© ISO 2019 – All rights reserved

```
REFERENCE FROM basic_attribute_schema
  (get_id_value,
   id_attribute,
   id_attribute_select);

REFERENCE FROM process_property_schema
  (action_property,
   resource_property);

REFERENCE FROM product_definition_schema
  (product_definition,
   product_definition_relationship,
   product_definition_relationship_relationship);

REFERENCE FROM product_property_representation_schema
  (item_identified_representation_usage);

REFERENCE FROM product_structure_schema
  (assembly_component_usage,
   multi_level_reference_designator,
   next_assembly_usage_occurrence,
   product_definition_occurrence);

REFERENCE FROM shape_dimension_schema
  (dimensional_location,
   dimensional_size);

REFERENCE FROM shape_tolerance_schema
  (geometric_tolerance);

REFERENCE FROM support_resource_schema
  (bag_to_set,
   identifier,
   label,
   text);
```

23.2 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.

 © ISO 2019 – All rights reserved

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.

23.3 product_property_definition_schema type definitions

23.3.1 characterized_definition

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

EXPRESS specification:

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

23.3.2 characterized_item

The **characterized_item** type allows for the designation of an item_identified_representation_usage, a dimensional_size, or a geometric_tolerance to which properties may be assigned.

EXPRESS specification:

```
TYPE characterized_item = SELECT
  (item_identified_representation_usage,
   dimensional_size,
   geometric_tolerance);
END_TYPE;
```

23.3.3 ppds_id_attribute_select

EXPRESS specification:

```
TYPE ppds_id_attribute_select = SELECT BASED_ON id_attribute_select WITH
  (property_definition,
   shape_aspect,
   shape_aspect_relationship);
END_TYPE;
```

23.3.4 characterized_product_definition

The **characterized_product_definition** type allows for the designation of a product_definition, or a product_definition_relationship, or a product_definition_relationship_relationship, or a product_definition_occurrence 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. Selection of

© ISO 2019 – All rights reserved

a product_definition_relationship_relationship means that the property is applied to the related attribute in the context of its relating 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_occurrence,
   product_definition_relationship,
   product_definition_relationship_relationship);
END_TYPE;
```

23.3.5 derived_property_select

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

EXPRESS specification:

```
TYPE derived_property_select = SELECT
  (property_definition,
   action_property,
   resource_property,
   dimensional_location,
   dimensional_size,
   geometric_tolerance);
END_TYPE;
```

23.3.6 shape_aspect_or_characterized_object

EXPRESS specification:

```
TYPE shape_aspect_or_characterized_object = SELECT
  (shape_aspect,
   characterized_object);
END_TYPE;
```

23.3.7 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_aspect.

EXPRESS specification:

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

23.3.8 internal_or_reflected_shape_aspect

EXPRESS specification:

```
TYPE internal_or_reflected_shape_aspect = SELECT
  (shape_aspect,
```

 © ISO 2019 – All rights reserved

```
   identifier);
END_TYPE;
```

23.3.9 multi_or_next_assembly_usage_occurrence

EXPRESS specification:

```
TYPE multi_or_next_assembly_usage_occurrence = SELECT
  (multi_level_reference_designator,
   next_assembly_usage_occurrence);
END_TYPE;
```

23.4 product_property_definition_schema entity definitions

23.4.1 characterized_object

A **characterized_object** is the identification of an item 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;
```

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.

23.4.2 characterized_object_relationship

A **characterized_object_relationship** is a relationship between 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;
```

© ISO 2019 – All rights reserved

```
END_ENTITY;
```

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, 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.

23.4.3 component_path_shape_aspect

A **component_path_shape_aspect** is a type of shape_aspect that identifies a portion of a shape of an assembly by referencing a shape_aspect that represents a portion of the shape of an occurrence of a part within this assembly. The referenced part and its shape_aspect might be defined locally or externally.

EXPRESS specification:

```
ENTITY component_path_shape_aspect
  SUBTYPE OF (shape_aspect);
  location : multi_or_next_assembly_usage_occurrence;
  component_shape_aspect : internal_or_reflected_shape_aspect;
UNIQUE
  UR1: location, component_shape_aspect;
END_ENTITY;
```

Attribute definitions:

location the occurrence of a part by its next_assembly_usage_occurrence or multi_level_reference_designator.

component_shape_aspect the locally or externally defined shape_aspect. In the case that it is externally defined, only the id of the externally defined shape_aspect is provided.

23.4.4 constituent_shape_aspect

A **constituent_shape_aspect** is a type of shape_aspect that identifies a portion of a shape_aspect.

EXPRESS specification:

```
ENTITY constituent_shape_aspect
  SUBTYPE OF (shape_aspect);
  parent : shape_aspect;
DERIVE
  SELF\shape_aspect.of_shape : product_definition_shape := get_root_shape_aspect(SELF);
END_ENTITY;
```


© ISO 2019 – All rights reserved

Attribute definitions:

parent the shape_aspect of which this entity is an aspect.

of_shape the product_definition_shape of which this entity is an aspect.

23.4.5 general_property

A **general_property** is an identification of 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;
```

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.

23.4.6 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;
```

© ISO 2019 – All rights reserved

```
  base_definition : general_property;
  derived_definition : derived_property_select;
WHERE
  WR1: (SIZEOF(['SHAPE_DIMENSION_SCHEMA.DIMENSIONAL_LOCATION', 'SHAPE_DIMENSION_SCHEMA.
DIMENSIONAL_SIZE', 'SHAPE_TOLERANCE_SCHEMA.GEOMETRIC_TOLERANCE'] * TYPEOF(derived_
definition)) > 0) OR (SIZEOF(USEDIN(derived_definition, ('PRODUCT_PROPERTY_DEFINITION_
SCHEMA.' + 'GENERAL_PROPERTY_ASSOCIATION.') + 'DERIVED_DEFINITION')) = 1);
  WR2: (SIZEOF(['SHAPE_DIMENSION_SCHEMA.DIMENSIONAL_LOCATION', 'SHAPE_DIMENSION_SCHEMA.
DIMENSIONAL_SIZE', 'SHAPE_TOLERANCE_SCHEMA.GEOMETRIC_TOLERANCE'] * TYPEOF(derived_
definition)) > 0) OR (derived_definition.name = base_definition.name);
END_ENTITY;
```

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.

Attribute definitions:

WR1 In the case that the **derived_definition** is of type property_definition, action_property or resource_property it shall be referred to by at most one **general_property_association**.

WR2 In the case that the **derived_definition** is of type property_definition, action_property or resource_property the attribute name of the **derived_definition** shall be the same as the attribute name of the **base_definition**.

23.4.7 general_property_relationship

A **general_property_relationship** is a relationship between 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;
```

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.

 © ISO 2019 – All rights reserved

relating_property one of the instances 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.

23.4.8 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;
```

Attribute definitions:

WR1 The definition attribute shall not refer to a shape_definition.

23.4.9 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;
```

© ISO 2019 – All rights reserved

```
  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;
```

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**.

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.

Attribute definitions:

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

NOTE 6 The id_attribute data type is defined in support_resource_schema of this part of ISO 10303.

NOTE 7 A template for constraining the population of the entity data types defined in the support_resource_schema is described in annex E.

23.4.10 shape_aspect

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

NOTE 1 A **shape_aspect** that is part of another **shape_aspect** allows decomposition of a **shape_aspect**.

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
  SUPERTYPE OF (ONEOF(shape_aspect_occurrence, component_path_shape_aspect) ANDOR
constituent_shape_aspect);
  name : label;
  description : OPTIONAL text;
  of_shape : product_definition_shape;
  product_definitional : LOGICAL;
DERIVE
  id : identifier := get_id_value(SELF);
```


© ISO 2019 – All rights reserved

```
UNIQUE
  UR1: id, of_shape;
WHERE
  WR1: SIZEOF(USEDIN(SELF, 'BASIC_ATTRIBUTE_SCHEMA.' + 'ID_ATTRIBUTE.IDENTIFIED_ITEM'))
<= 1;
END_ENTITY;
```

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**.

NOTE 2 This attribute is an upwardly compatible addition to **shape_aspect** 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.

Attribute definitions:

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

NOTE 5 The id_attribute data type is defined in basic_attribute_schema 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.

23.4.11 shape_aspect_occurrence

A **shape_aspect_occurrence** is a type of shape_aspect that is defined by another shape_aspect or a characterized_object.

EXPRESS specification:

```
ENTITY shape_aspect_occurrence
```

© ISO 2019 – All rights reserved

```
  SUBTYPE OF (shape_aspect);
  definition : shape_aspect_or_characterized_object;
WHERE
  WR1: acyclic_shape_aspect_occurrence(SELF, definition);
END_ENTITY;
```

Attribute definitions:

definition the shape_aspect or characterized_object that is defining this **shape_aspect_occurrence**.

Attribute definitions:

WR1 In the case that a **shape_aspect_occurrence** is defined by another **shape_aspect_occurrence**, the graph established by the attribute definition shall not be cyclic.

23.4.12 shape_aspect_relationship

A **shape_aspect_relationship** is a relationship between 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;
```

Attribute definitions:

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

© ISO 2019 – All rights reserved

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 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.

id the identifier that distinguishes the **shape_aspect_relationship**.

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.

Attribute definitions:

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 basic_attribute_schema 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.

23.5 product_property_definition_schema function definitions

23.5.1 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.

© ISO 2019 – All rights reserved

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.relating_object IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(ca <* bag_to_set(USEDIN(relation.relating_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.relating_
object, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

23.5.1.1 Parameters

23.5.1.1.1 relation

(input) the candidate characterized_object_relationship to be checked.

EXPRESS specification:

```
relation : characterized_object_relationship
```

23.5.1.1.2 relatives

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

EXPRESS specification:

```
relatives : SET [1:?] OF characterized_object
```

23.5.1.1.3 specific_relation

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

EXPRESS specification:

```
specific_relation : STRING
```

23.5.1.2 Variables

23.5.1.2.1 x

EXPRESS specification:

```
x : SET OF characterized_object_relationship;
```

 © ISO 2019 – All rights reserved

23.5.1.3 Statements

23.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_characterized_object_relationship(x[i], relatives + relation.relating_
object, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

23.5.2 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.relating_property IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(genp <* bag_to_set(USEDIN(relation.relating_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.relating_
property, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

23.5.2.1 Parameters

23.5.2.1.1 relation

(input) the candidate general_property_relationship to be checked.

EXPRESS specification:

```
relation : general_property_relationship
```

23.5.2.1.2 relatives

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

© ISO 2019 – All rights reserved

EXPRESS specification:

```
relatives : SET [1:?] OF general_property
```

23.5.2.1.3 specific_relation

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

EXPRESS specification:

```
specific_relation : STRING
```

23.5.2.2 Variables

23.5.2.2.1 x

EXPRESS specification:

```
x : SET OF general_property_relationship;
```

23.5.2.3 Statements

23.5.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_general_property_relationship(x[i], relatives + relation.relating_
property, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

23.5.3 acyclic_shape_aspect_occurrence

The **acyclic_shape_aspect_occurrence** function determines whether the graph of instances of the entity data type shape_aspect_occurrence that are linked by the attribute definition contains a cycle.

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_shape_aspect_occurrence(sao : shape_aspect_occurrence;
                                         definition : shape_aspect_or_characterized_
object) : BOOLEAN;
  IF NOT ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_ASPECT_OCCURRENCE' IN
TYPEOF(definition)) THEN
    RETURN (TRUE);
  END_IF;
  IF definition :=: sao THEN
    RETURN (FALSE);
  ELSE
    RETURN (acyclic_shape_aspect_occurrence(sao, definition\shape_aspect_occurrence.
definition));
  END_IF;
END_FUNCTION;
```

23.5.3.1 Parameters

23.5.3.1.1 sao

(input) the candidate shape_aspect_occurrence to be checked.

© ISO 2019 – All rights reserved

EXPRESS specification:

```
sao : shape_aspect_occurrence
```

23.5.3.1.2 definition

(input) the candidate shape_aspect_or_characterized_object to be checked.

EXPRESS specification:

```
definition : shape_aspect_or_characterized_object
```

23.5.4 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.relating_shape_aspect IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(sa <* bag_to_set(USEDIN(relation.relating_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.relating_shape_
aspect, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

23.5.4.1 Parameters

23.5.4.1.1 relation

(input) the candidate shape_aspect_relationship to be checked.

EXPRESS specification:

```
relation : shape_aspect_relationship
```

23.5.4.1.2 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.

EXPRESS specification:

© ISO 2019 – All rights reserved

```
relatives : SET [1:?] OF shape_aspect
```

23.5.4.1.3 specific_relation

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

EXPRESS specification:

```
specific_relation : STRING
```

23.5.4.2 Variables

23.5.4.2.1 x

EXPRESS specification:

```
x : SET OF shape_aspect_relationship;
```

23.5.4.3 Statements

23.5.4.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_shape_aspect_relationship(x[i], relatives + relation.relating_shape_
aspect, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

23.5.5 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;
  REPEAT i := 1 TO HIINDEX(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;
```


© ISO 2019 – All rights reserved

23.5.5.1 Parameters

23.5.5.1.1 c_def_instance

(input) the candidate characterized_definition to be checked.

EXPRESS specification:

```
c_def_instance : characterized_definition
```

23.5.5.2 Variables

23.5.5.2.1 pd_set

EXPRESS specification:

```
pd_set : SET OF product_definition_shape := [];
```

23.5.5.2.2 pdr_set

EXPRESS specification:

```
pdr_set : SET OF shape_aspect := [];
```

23.5.5.3 Statements

23.5.5.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(pd_set);
  pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i], 'PRODUCT_PROPERTY_DEFINITION_SCHEMA.
SHAPE_ASPECT.OF_SHAPE'));
END_REPEAT;
```

23.5.6 get_root_shape_aspect

The **get_root_shape_aspect** function returns the product_definition_shape of which the current shape_aspect is an identified aspect.

EXPRESS specification:

```
FUNCTION get_root_shape_aspect(current : shape_aspect) : product_definition_shape;
  REPEAT WHILE 'PRODUCT_DEFINITION_SCHEMA.CONSTITUENT_SHAPE_ASPECT' IN TYPEOF(current);
    current := current.parent;
  END_REPEAT;
  RETURN (current.of_shape);
END_FUNCTION;
```

23.5.6.1 Parameters

23.5.6.1.1 current

(input) the shape_aspect that is the basis for the function.

EXPRESS specification:

```
current : shape_aspect
```

23.5.6.2 Statements

23.5.6.2.1 1

EXPRESS specification:

```
REPEAT WHILE 'PRODUCT_DEFINITION_SCHEMA.CONSTITUENT_SHAPE_ASPECT' IN TYPEOF(current);
```

© ISO 2019 – All rights reserved

```
  current := current.parent;
END_REPEAT;
```

24 product_property_representation_schema

24.1 General

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**.

EXPRESS specification:

```
SCHEMA product_property_representation_schema '{iso standard 10303 part(41) version(9)
object(1) product_property_representation_schema(20)}';

REFERENCE FROM basic_attribute_schema
  (description_attribute,
   description_attribute_select,
   get_description_value,
   get_name_value,
   name_attribute,
   name_attribute_select);

REFERENCE FROM material_property_definition_schema
  (property_definition_relationship);

REFERENCE FROM product_definition_schema
  (product_definition,
   product_definition_relationship);

REFERENCE FROM product_property_definition_schema
  (characterized_object,
   characterized_definition,
   general_property,
   product_definition_shape,
   property_definition,
   shape_aspect,
   shape_aspect_relationship);

REFERENCE FROM product_structure_schema
  (product_definition_specified_occurrence);

REFERENCE FROM representation_schema
  (representation,
   representation_item,
   representation_relationship,
   representation_reference,
   using_representations,
   mapped_item,
   representation_context,
   representation_map,
   list_representation_item,
   set_representation_item,
   get_representations_for_items);

REFERENCE FROM support_resource_schema
```

 © ISO 2019 – All rights reserved

```
  (bag_to_set,
   label,
   text);
```

24.2 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.

24.3 product_property_representation_schema type definitions

24.3.1 pprs_description_attribute_select

EXPRESS specification:

```
TYPE pprs_description_attribute_select = SELECT BASED_ON description_attribute_select WITH
  (context_dependent_shape_representation,
   property_definition_representation);
END_TYPE;
```

24.3.2 pprs_name_attribute_select

EXPRESS specification:

```
TYPE pprs_name_attribute_select = SELECT BASED_ON name_attribute_select WITH
  (context_dependent_shape_representation,
   property_definition_representation);
END_TYPE;
```

24.3.3 chained_representation_link

The **chained_representation_link** type allows for the designation of a mapped_item, a representation_context, or a representation_relationship.

EXPRESS specification:

```
TYPE chained_representation_link = SELECT
  (mapped_item,
   representation_context,
   representation_relationship);
END_TYPE;
```

24.3.4 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:

© ISO 2019 – All rights reserved

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

24.3.5 item_identified_representation_usage_definition

EXPRESS specification:

```
TYPE item_identified_representation_usage_definition = EXTENSIBLE GENERIC_ENTITY SELECT
  (represented_definition);
END_TYPE;
```

24.3.6 item_identified_representation_usage_select

EXPRESS specification:

```
TYPE item_identified_representation_usage_select = SELECT
  (representation_item,
   list_representation_item,
   set_representation_item);
END_TYPE;
```

24.4 product_property_representation_schema entity definitions

24.4.1 chain_based_item_identified_representation_usage

A **chain_based_item_identified_representation_usage** is a type of item_identified_representation_usage that represents a chain of representations in a graph of representations, where the undirected links in the graph can be instances of entity data types representation_context, representation_relationship, or mapped_item. The directed links are derived from the nodes and undirected links and consist solely of instances of the entity data type representation_relationship. The attribute rep_2 of the derived instances is directed towards the root, while the attribute rep_1 is directed towards the leaf.

EXPRESS specification:

```
ENTITY chain_based_item_identified_representation_usage
  SUBTYPE OF (item_identified_representation_usage);
  nodes : LIST [2:?] OF UNIQUE representation;
  undirected_link : LIST [1:?] OF chained_representation_link;
DERIVE
  root : representation := nodes[1];
  SELF\item_identified_representation_usage.used_representation RENAMED leaf :
representation := nodes[HIINDEX(nodes)];
  directed_link : LIST [1:?] OF representation_relationship := get_directed_link(nodes,
undirected_link);
WHERE
  WR1: EXISTS(directed_link);
END_ENTITY;
```

Attribute definitions:

nodes the list of representation in the chain.

undirected_link the list of items that relate nodes in the chain.

© ISO 2019 – All rights reserved

NOTE 1 The items in the **undirected_link** are placed correctly relative to the nodes in the chain but are not guaranteed to have a consistent direction along the chain.

root the node that is the initial terminus of the chain.

leaf the node that is the final terminus of the chain.

directed_link the list of representation_relationship that are derived from the **undirected_link** and that are consistently directed from the root to the leaf.

Attribute definitions:

WR1 The directed_link attribute shall be valid for the given set of nodes and undirected_links.

NOTE 2 the get_directed_link function returns indeterminate if it cannot construct a valid list. The EXPRESS EXISTS function returns FALSE with an indeterminate input.

24.4.2 characterized_chain_based_item_within_representation

A **characterized_chain_based_item_within_representation** is a type of characterized_item_within_representation that represents a chain of representations in a graph of representations, where the undirected links in the graph can be instances of entity data types representation_context, representation_relationship, or mapped_item. The directed links are derived from the nodes and undirected links and consist solely of instances of the entity data type representation_relationship. The attribute rep_2 of the derived instances is directed towards the root, while the attribute rep_1 is directed towards the leaf.

EXPRESS specification:

```
ENTITY characterized_chain_based_item_within_representation
  SUBTYPE OF (characterized_item_within_representation);
  nodes : LIST [2:?] OF UNIQUE representation;
  undirected_link : LIST [1:?] OF chained_representation_link;
DERIVE
  root : representation := nodes[1];
  SELF\characterized_item_within_representation.rep RENAMED leaf : representation :=
nodes[HIINDEX(nodes)];
  directed_link : LIST [1:?] OF representation_relationship := get_directed_link(nodes,
undirected_link);
WHERE
  WR1: EXISTS(directed_link);
END_ENTITY;
```

Attribute definitions:

nodes the list of representation in the chain.

undirected_link the list of items that relate nodes in the chain.

NOTE 1 The items in the **undirected_link** are placed correctly relative to the nodes in the chain but are not guaranteed to have a consistent direction along the chain.

root the node that is the initial terminus of the chain.

leaf the node that is the final terminus of the chain.

© ISO 2019 – All rights reserved

directed_link the list of representation_relationship that are derived from the **undirected_link** and that are consistently directed from the root to the leaf.

Attribute definitions:

WR1 The directed_link attribute shall be valid for the given set of nodes and undirected_links.

NOTE 2 the get_directed_link function returns indeterminate if it cannot construct a valid list. The EXPRESS EXISTS function returns FALSE with an indeterminate input.

24.4.3 characterized_item_within_representation

A **characterized_item_within_representation** is a type of characterized_object that provides the characterization of a representation_item within a specified representation.

EXPRESS specification:

```
ENTITY characterized_item_within_representation
  SUBTYPE OF (characterized_object);
  item : representation_item;
  rep : representation;
UNIQUE
  UR1: item, rep;
WHERE
  WR1: rep IN using_representations(item);
END_ENTITY;
```

Attribute definitions:

item the representation_item that is characterised.

rep the representation in which the characterization of the representation_item applies.

Attribute definitions:

WR1 The **item** shall be founded in the specified representation.

24.4.4 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
```

 © ISO 2019 – All rights reserved

```
  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;
```

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**.

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.

Attribute definitions:

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 basic_attribute_schema 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 basic_attribute_schema 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.

24.4.5 item_identified_representation_usage

An **item_identified_representation_usage** is an identification of a single or a set or a list of representation_item(s) within a representation as being the element(s) that describes a particular component or part of the property that is described by the representation. More than one representation. may be bound to a product data item by members of **item_identified_representation_usage** and more than one member of representation_item(s) may be bound to a product data item by members of **item_identified_representation_usage**. The same item of product data shall not be bound to more than one member of representation_item within a member of representation.

© ISO 2019 – All rights reserved

EXAMPLE 1 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 **shape_aspect**, a hole, in the product. **Item_identification_representation_usage** is used to state that the curve referenced by 'identified_item' is a representation item for the **shape_aspect** referenced by 'definition' and that the whole representation is referenced by 'used_representation' to indicate the structural context for the curve. **Geometric_representation_context** plays no role in this example.

EXAMPLE 2 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 **shape_aspect**, a hole, in the product. A member of **Item_identification_representation_usage** is used to state that the curve referenced by 'identified_item' is a representation item for the **shape_aspect#1** referenced by 'definition' and that the whole representation is referenced by 'used_representation' to indicate the structural context for the curve. A second member of **Item_identification_representation_usage** references a different curve in that same representation and references the same **shape_aspect#1**. The data population is invalid because only one item within a representation can be bound to a **shape_aspect**.

EXPRESS specification:

```
ENTITY item_identified_representation_usage;
  name : label;
  description : OPTIONAL text;
  definition : item_identified_representation_usage_definition;
  used_representation : representation;
  identified_item : item_identified_representation_usage_select;
UNIQUE
  UR1: used_representation, identified_item;
  UR2: used_representation, definition;
WHERE
  WR1: NOT (SIZEOF(identified_item) = 1) OR (used_representation IN using_
representations(identified_item));
  WR2: NOT (SIZEOF(identified_item) > 1) OR (used_representation IN get_representations_
for_items(identified_item));
END_ENTITY;
```

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 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 or list_representation_item or set_representation_item that describes the identified property or aspect.

Attribute definitions:

WR1 If the size of identified_item is one then the used_representation shall reference the item directly or indirectly.

WR2 If the size of identified_item is greater than one then the used_representation shall be one of the representations that reference the item directly or indirectly.

© ISO 2019 – All rights reserved

24.4.6 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;
```

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**.

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.

Attribute definitions:

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 basic_attribute_schema 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 basic_attribute_schema 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.

© ISO 2019 – All rights reserved

24.4.7 shape_definition_representation

A **shape_definition_representation** is a type of property_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);
  SELF\property_definition_representation.definition : property_definition;
  SELF\property_definition_representation.used_representation : shape_representation;
WHERE
  WR1: ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
TYPEOF(definition)) OR ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_DEFINITION' IN
TYPEOF(definition\property_definition.definition));
END_ENTITY;
```

Attribute definitions:

definition an inherited attribute whose value shall be of type property_definition

used_representation an inherited attribute whose value shall be of type shape_representation.

Attribute definitions:

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

24.4.8 shape_representation

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

EXPRESS specification:

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

24.4.9 shape_representation_reference

A **shape_representation_reference** is a type of representation_reference in which the referenced data type is a shape_representation.

EXPRESS specification:

```
ENTITY shape_representation_reference
  SUBTYPE OF (representation_reference);
END_ENTITY;
```

24.4.10 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 or shape_representation_reference.

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);
```


© ISO 2019 – All rights reserved

```
WHERE
  WR1: SIZEOF(['PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_REPRESENTATION',
'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_REPRESENTATION_REFERENCE'] * (TYPEOF(SELF
\representation_relationship.rep_1) + TYPEOF(SELF\representation_relationship.rep_2))) >=
1;
END_ENTITY;
```

Attribute definitions:

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

24.4.11 specified_occurrence_context_dependent_shape_representation

A **specified_occurrence_context_dependent_shape_representation** is a type of context_dependent_shape_representation that established a relation for a hierarchical product structure using product_definition_specified_occurrence.

EXPRESS specification:

```
ENTITY specified_occurrence_context_dependent_shape_representation
  SUBTYPE OF (context_dependent_shape_representation);
  sub_element : product_definition_specified_occurrence;
WHERE
  WR1: 'PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION_OCCURRENCE' IN TYPEOF(SELF.
represented_product_relation.definition.related_product_definition);
  WR2: sub_element IN represented_product_relation.definition.related_product_definition.
descendant_occurrences;
END_ENTITY;
```

Attribute definitions:

sub_element the product_definition_specified_occurrence for which the association established by the inherited attributes **representation_relation** and **represented_product_relation** applies.

Attribute definitions:

WR1 The related_product_definition of the **represented_product_relation** shall be of type product_definition_occurrence.

WR2 The **sub_element** shall be within the descendant_occurrences of the product_definition_occurrence that is referenced by the related_product_definition of the **represented_product_relation**.

24.5 product_property_representation_schema function definitions

24.5.1 get_directed_link

The **get_directed_link** function returns for any list of representation and associated list of chained_representation_link the resolved list of representation_relationship when each member of **undirected_link** references the correct members of **nodes**. The **get_directed_link** function returns UNKNOWN when the input does not satisfy the correctness requirement.

EXPRESS specification:

```
FUNCTION get_directed_link(nodes : LIST OF representation;
                           undirected_link : LIST OF chained_representation_link) : LIST
OF representation_relationship;
  LOCAL
```

© ISO 2019 – All rights reserved

```
    directed_link : LIST OF representation_relationship := [];
  END_LOCAL;
  IF (SIZEOF(nodes) <> SIZEOF(undirected_link) + 1) OR (value_unique(nodes) = FALSE) THEN
    RETURN (?);
  END_IF;
  REPEAT i := 1 TO SIZEOF(undirected_link);
    CASE TRUE OF
    'REPRESENTATION_SCHEMA.REPRESENTATION_CONTEXT' IN TYPEOF(undirected_link[i]) :
      BEGIN
        IF (nodes[i]\representation.context_of_items :=: undirected_link[i]) AND (nodes[i
+ 1]\representation.context_of_items :=: undirected_link[i]) THEN
          INSERT(directed_link, representation_relationship('', '', nodes[i + 1],
nodes[i]), i - 1);
        ELSE
          RETURN (?);
        END_IF;
      END;
    'REPRESENTATION_SCHEMA.REPRESENTATION_RELATIONSHIP' IN TYPEOF(undirected_link[i]) :
      BEGIN
        IF (nodes[i] :=: undirected_link[i]\representation_relationship.rep_1) AND
(nodes[i + 1] :=: undirected_link[i]\representation_relationship.rep_2) OR (nodes[i] :=
: undirected_link[i]\representation_relationship.rep_2) AND (nodes[i + 1] :=: undirected_
link[i]\representation_relationship.rep_1) THEN
          INSERT(directed_link, representation_relationship('', '', nodes[i + 1],
nodes[i]), i - 1);
        ELSE
          RETURN (?);
        END_IF;
      END;
    'REPRESENTATION_SCHEMA.MAPPED_ITEM' IN TYPEOF(undirected_link[i]) :
      BEGIN
        IF (nodes[i] IN using_representations(undirected_link[i])) AND (nodes[i + 1] :=:
undirected_link[i]\mapped_item.mapping_source\representation_map.mapped_representation)
THEN
          INSERT(directed_link, representation_relationship('', '', nodes[i + 1],
nodes[i]), i - 1);
        ELSE
          RETURN (?);
        END_IF;
      END;
    OTHERWISE :
      RETURN (?);
    END_CASE;
  END_REPEAT;
  IF value_unique(directed_link) = FALSE THEN
    RETURN (?);
  END_IF;
  RETURN (directed_link);
END_FUNCTION;
```

24.5.1.1 Parameters

24.5.1.1.1 nodes

(input) the list of instances of the entity data type representation to be used as a reference for constructing the directed_link.

EXPRESS specification:

```
nodes : LIST OF representation
```

 © ISO 2019 – All rights reserved

24.5.1.1.2 undirected_link

(input) the list of instances of the entity data type mapped_item, representation_context, or representation_relationship that the function is comparing against the reference to establish the directed_link.

EXPRESS specification:

```
undirected_link : LIST OF chained_representation_link
```

24.5.1.2 Variables

24.5.1.2.1 directed_link

EXPRESS specification:

```
directed_link : LIST OF representation_relationship := [];
```

24.5.1.3 Statements

24.5.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO SIZEOF(undirected_link);
  CASE TRUE OF
  'REPRESENTATION_SCHEMA.REPRESENTATION_CONTEXT' IN TYPEOF(undirected_link[i]) :
    BEGIN
      IF (nodes[i]\representation.context_of_items :=: undirected_link[i]) AND (nodes[i +
1]\representation.context_of_items :=: undirected_link[i]) THEN
        INSERT(directed_link, representation_relationship('', '', nodes[i + 1],
nodes[i]), i - 1);
      ELSE
        RETURN (?);
      END_IF;
    END;
  'REPRESENTATION_SCHEMA.REPRESENTATION_RELATIONSHIP' IN TYPEOF(undirected_link[i]) :
    BEGIN
      IF (nodes[i] :=: undirected_link[i]\representation_relationship.rep_1) AND
(nodes[i + 1] :=: undirected_link[i]\representation_relationship.rep_2) OR (nodes[i] :=
: undirected_link[i]\representation_relationship.rep_2) AND (nodes[i + 1] :=: undirected_
link[i]\representation_relationship.rep_1) THEN
        INSERT(directed_link, representation_relationship('', '', nodes[i + 1],
nodes[i]), i - 1);
      ELSE
        RETURN (?);
      END_IF;
    END;
  'REPRESENTATION_SCHEMA.MAPPED_ITEM' IN TYPEOF(undirected_link[i]) :
    BEGIN
      IF (nodes[i] IN using_representations(undirected_link[i])) AND (nodes[i + 1] :=:
undirected_link[i]\mapped_item.mapping_source\representation_map.mapped_representation)
THEN
        INSERT(directed_link, representation_relationship('', '', nodes[i + 1],
nodes[i]), i - 1);
      ELSE
        RETURN (?);
      END_IF;
    END;
  OTHERWISE :
    RETURN (?);
  END_CASE;
END_REPEAT;
```

© ISO 2019 – All rights reserved

24.5.2 relatives_of_product_definitions

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;
  RETURN (local_relatives_of_product_definitions(definition_set, definition_set, relation_
subtype));
END_FUNCTION;
```

24.5.2.1 Parameters

24.5.2.1.1 definition_set

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

EXPRESS specification:

```
definition_set : SET OF product_definition
```

24.5.2.1.2 relation_subtype

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

EXPRESS specification:

```
relation_subtype : STRING
```


© ISO 2019 – All rights reserved

24.5.3 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_representation_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;
```

24.5.3.1 Parameters

24.5.3.1.1 shape_representation_set

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

EXPRESS specification:

```
shape_representation_set : SET OF shape_representation
```

24.5.4 get_property_definition_representations

The **get_property_definition_representations** function returns for any characterized_definition the set of property_definition_representation objects 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;
```

© ISO 2019 – All rights reserved

```
  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;
  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;
```

24.5.4.1 Parameters

24.5.4.1.1 c_def_instance

(input) the candidate characterized_definition to be checked.

EXPRESS specification:

```
c_def_instance : characterized_definition
```

24.5.4.2 Variables

24.5.4.2.1 pd_set

EXPRESS specification:

```
pd_set : SET OF property_definition := [];
```

24.5.4.2.2 pdr_set

EXPRESS specification:

```
pdr_set : SET OF property_definition_representation := [];
```

24.5.4.3 Statements

24.5.4.3.1 i

EXPRESS specification:

```
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;
```

25 qualifications_schema

25.1 General

The subject of the **qualifications_schema** is the identification of types of qualifications and the relationships between them. This schema provides for the definition of generic data relating to types of qualifications.

EXAMPLE The identification of types of qualifications and the definition of various kinds of relationship between types of qualifications are examples of generic data relating to types of qualification.

EXPRESS specification:

 © ISO 2019 – All rights reserved

```
SCHEMA qualifications_schema '{iso standard 10303 part(41) version(6) object(1)
qualifications_schema(21)}';

REFERENCE FROM support_resource_schema
  (identifier,
   label,
   text,
   bag_to_set);
```

25.2 Fundamental concepts and assumptions

This schema provides resources for the identification, naming, and description of:

— qualifications;

— types of qualification;

— relationships between types of qualification.

25.3 qualifications_schema entity definitions

25.3.1 qualification

A **qualification** is a particular instance of the formal identification of a capability or aptitude in a specific person or organization.

EXAMPLE 1 John Smith's driving licence.

EXAMPLE 2 Mary Brown's doctorate entitled "Datamodelling Using EXPRESS2".

NOTE A relationship between a qualification_type object and one or more **qualification** objects is established by the declaration of a qualification_type_assignment subtype containing a single attribute that references a set of a SELECT type that contains an qualification.

EXPRESS specification:

```
ENTITY qualification;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **qualification**.

name the label by which the **qualification** is known.

description the text that characterizes the **qualification**. The value of this attribute need not be specified.

25.3.2 qualification_relationship

A **qualification_relationship** is a relationship between two instances of the qualification entity data type and provides for the identification, naming, and description of that relationship.

EXPRESS specification:

```
ENTITY qualification_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_qualification : qualification;
```

© ISO 2019 – All rights reserved

```
  related_qualification : qualification;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **qualification_relationship**.

name the label by which the **qualification_relationship** is known

description the text that characterizes the **qualification_relationship**. The value of this attribute need not be specified.

relating_qualification one of the instances of qualification that is a part of the relationship.

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.

related_qualification the other instance of qualification 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 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.

25.3.3 qualification_type

An **qualification_type** is the formal identification of a capability or aptitude.

EXAMPLE 1 A driving licence.

EXAMPLE 2 A university degree.

NOTE A relationship between a **qualification_type** object and one or more qualification objects is established by the declaration of an qualification_type_assignment subtype containing a single attribute that references a set of a SELECT type that contains a qualification.

EXPRESS specification:

```
ENTITY qualification_type;
  id : identifier;
  name : label;
  description : OPTIONAL text;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **qualification_type**.

name the label by which the **qualification_type** is known.

description the text that characterizes the **qualification_type**. The value of this attribute need not be specified.

25.3.4 qualification_type_relationship

A **qualification_type_relationship** is a relationship between two instances of the qualification_type entity data type and provides for the identification, naming, and description of that relationship.

 © ISO 2019 – All rights reserved

EXPRESS specification:

```
ENTITY qualification_type_relationship;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  relating_qualification_type : qualification_type;
  related_qualification_type : qualification_type;
END_ENTITY;
```

Attribute definitions:

id the identifier that distinguishes the **qualification_type_relationship**.

name the label by which the **qualification_type_relationship** is known.

description the text that characterizes the **qualification_type_relationship**. The value of this attribute need not be specified.

relating_qualification_type one of the instances of qualification_type that is a part of the relationship.

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.

related_qualification_type the other instance of qualification_type 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 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.

25.4 qualifications_schema function definitions

25.4.1 acyclic_qualification_relationship

The **acyclic_qualification_relationship** function determines whether the graph of instances of the entity data type qualification that contains relation as one of its links contains a cycle. This function may be used to evaluate either an qualification_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 qualification_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_qualification_relationship(relation : qualification_relationship;
                                            relatives : SET OF qualification;
                                            specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF qualification_relationship;
  END_LOCAL;
  IF relation.relating_qualification IN relatives THEN
    RETURN (FALSE);
  END_IF;
```

© ISO 2019 – All rights reserved

```
  x := QUERY(qual <* bag_to_set(USEDIN(relation.relating_qualification, ('QUALIFICATIONS_
SCHEMA.' + 'QUALIFICATION_RELATIONSHIP.') + 'RELATED_QUALIFICATION')) | specific_relation
IN TYPEOF(qual));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_qualification_relationship(x[i], relatives + relation.relating_
qualification, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

25.4.1.1 Parameters

25.4.1.1.1 relation

(input) the candidate qualification_relationship to be checked.

EXPRESS specification:

```
relation : qualification_relationship
```

25.4.1.1.2 relatives

(input) the set of instances of the entity data type qualification that the function is searching for in the relating_qualification parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET OF qualification
```

25.4.1.1.3 specific_relation

(input) the fully qualified name of a subtype of the qualification_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

25.4.1.2 Variables

25.4.1.2.1 x

EXPRESS specification:

```
x : SET OF qualification_relationship;
```

25.4.1.3 Statements

25.4.1.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_qualification_relationship(x[i], relatives + relation.relating_
qualification, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```


© ISO 2019 – All rights reserved

25.4.2 acyclic_qualification_type_relationship

The **acyclic_qualification_type_relationship** function determines whether the graph of instances of the entity data type qualification_type that contains relation as one of its links contains a cycle. This function may be used to evaluate either a qualification_type_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 qualification_type_relationship entity include rules that use this function.

EXPRESS specification:

```
FUNCTION acyclic_qualification_type_relationship(relation : qualification_type_
relationship;
                                                 relatives : SET OF qualification_type;
                                                 specific_relation : STRING) : BOOLEAN;
  LOCAL
    x : SET OF qualification_type_relationship;
  END_LOCAL;
  IF relation.relating_qualification_type IN relatives THEN
    RETURN (FALSE);
  END_IF;
  x := QUERY(qultyp <* bag_to_set(USEDIN(relation.relating_qualification_type,
('QUALIFICATIONS_SCHEMA.' + 'QUALIFICATION_TYPE_RELATIONSHIP.') + 'RELATED_QUALIFICATION_
TYPE')) | specific_relation IN TYPEOF(qultyp));
  REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_qualification_type_relationship(x[i], relatives + relation.relating_
qualification_type, specific_relation) THEN
      RETURN (FALSE);
    END_IF;
  END_REPEAT;
  RETURN (TRUE);
END_FUNCTION;
```

25.4.2.1 Parameters

25.4.2.1.1 relation

(input) the candidate qualification_type_relationship to be checked.

EXPRESS specification:

```
relation : qualification_type_relationship
```

25.4.2.1.2 relatives

(input) the set of instances of the entity data type qualification_type that the function is searching for in the relating_qualification_type parameter of the **relation** argument.

EXPRESS specification:

```
relatives : SET OF qualification_type
```

25.4.2.1.3 specific_relation

(input) the fully qualified name of a subtype of the qualification_type_relationship entity.

EXPRESS specification:

```
specific_relation : STRING
```

© ISO 2019 – All rights reserved

25.4.2.2 Variables

25.4.2.2.1 x

EXPRESS specification:

```
x : SET OF qualification_type_relationship;
```

25.4.2.3 Statements

25.4.2.3.1 i

EXPRESS specification:

```
REPEAT i := 1 TO HIINDEX(x);
  IF NOT acyclic_qualification_type_relationship(x[i], relatives + relation.relating_
qualification_type, specific_relation) THEN
    RETURN (FALSE);
  END_IF;
END_REPEAT;
```

26 security_classification_schema

26.1 General

The subject of the **security_classification_schema** is the definition of degrees of secrecy to be specified.

EXPRESS specification:

```
SCHEMA security_classification_schema '{iso standard 10303 part(41) version(3) object(1)
security_classification_schema(22)}';

REFERENCE FROM support_resource_schema
  (label,
   text);
```

26.2 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.

26.3 security_classification_schema entity definitions

26.3.1 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;
```

Attribute definitions:

 © ISO 2019 – All rights reserved

name the label by which the **security_classification** is known.

EXAMPLE 'National Security' is a security classification role. A 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.

26.3.2 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 levels.

EXPRESS specification:

```
ENTITY security_classification_level;
  name : label;
END_ENTITY;
```

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.

27 support_resource_schema

27.1 General

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.

EXPRESS specification:

```
SCHEMA support_resource_schema '{iso standard 10303 part(41) version(7) object(1) support_
resource_schema(23)}';
```

27.2 Fundamental concepts and assumptions

An identifier is a means that enables a processor to distinguish an information from other information of the same kind. A label is textual information assigned to an item of information in order to facilitate its access by a human being.

Product data items 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.

© ISO 2019 – All rights reserved

Product data items 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.

27.3 support_resource_schema type definitions

27.3.1 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;
```

27.3.2 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;
```

27.3.3 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;
```

27.4 support_resource_schema function definitions

27.4.1 bag_to_set

The **bag_to_set** 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;
```

 © ISO 2019 – All rights reserved

27.4.1.1 Parameters

27.4.1.1.1 the_bag

(input) the BAG that is to be converted into a SET.

EXPRESS specification:

```
the_bag : BAG OF GENERIC : intype
```

27.4.1.2 Variables

27.4.1.2.1 the_set

EXPRESS specification:

```
the_set : SET OF GENERIC : intype := [];
```

27.4.2 type_check_function

The **type_check_function** 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;
  IF NOT EXISTS(the_type) 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 :
```

© ISO 2019 – All rights reserved

```
      RETURN (SIZEOF(sub_names * TYPEOF(the_type)) <= 1);
    OTHERWISE :
      RETURN (UNKNOWN);
    END_CASE;
  END_IF;
END_FUNCTION;
```

27.4.2.1 Parameters

27.4.2.1.1 the_type

(input) the instance whose data type is checked.

EXPRESS specification:

```
the_type : GENERIC
```

27.4.2.1.2 sub_names

(input) the set of names of the entity data types that are candidates for the type of **the_type**.

EXPRESS specification:

```
sub_names : SET OF STRING
```

27.4.2.1.3 criterion

(input) the integer number value between 0 and 3.

EXPRESS specification:

```
criterion : INTEGER
```


© ISO 2019 – All rights reserved

Bibliography

© ISO 2019 – All rights reserved

Descriptors: integrated generic resource, product definition, product property, measure, application context, product property representation, person, organization, experience, qualifications, location and position in organization.

Price based on 271 pages

© ISO 2019 – All rights reserved