

---

---

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

Part 212:

**Application protocol: Electrotechnical  
design and installation**

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

*Partie 212: Protocole d'application: Conception électrotechnique et  
installation*

**PDF disclaimer**

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

Adobe is a trademark of Adobe Systems Incorporated.

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

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-212:2001

© ISO 2001

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.ch](mailto:copyright@iso.ch)  
Web [www.iso.ch](http://www.iso.ch)

Printed in Switzerland

## Contents

	Page
1 Scope.....	1
2 Normative references .....	3
3 Terms, definitions, and abbreviations.....	5
3.1 Terms defined in IEC 61082-1 .....	5
3.2 Terms defined in ISO 10303-1 .....	5
3.3 Terms defined in ISO 10303-31 .....	6
3.4 Terms defined in ISO 10303-42 .....	6
3.5 Terms defined in ISO 10303-44 .....	6
3.6 Terms defined in ISO 10303-46 .....	6
3.7 Other terms and definitions .....	6
3.8 Abbreviations .....	10
4 Information requirements.....	11
4.1 Units of functionality .....	11
4.1.1 allocation (AL1).....	12
4.1.2 classification (CA1).....	12
4.1.3 conditions (CD1) .....	13
4.1.4 configuration_management (CF1).....	13
4.1.5 course (CO1).....	14
4.1.6 designation (DE1).....	15
4.1.7 dimensioned_documentation (DI1) .....	15
4.1.8 documentation (DO1) .....	16
4.1.9 effectivity_data (EF1).....	19
4.1.10 external_reference (ER1).....	19
4.1.11 function_structure (F1).....	21
4.1.12 functional_connectivity (FC1).....	21

## ISO 10303-212:2001(E)

4.1.13	installation (IN1)	22
4.1.14	messages (M1)	24
4.1.15	network_allocation (NA1)	24
4.1.16	organizational_data (OD1)	25
4.1.17	physical_connectivity (PC1)	25
4.1.18	product_structure (PD1)	27
4.1.19	properties (PR1)	28
4.1.20	remark (R1)	30
4.1.21	schematic_documentation (SC1)	30
4.1.22	site (SI1)	31
4.1.23	work_management (W1)	31
4.2	Application objects	32
4.3	Application assertions	403
5	Application interpreted model	664
5.1	Mapping table	664
5.2	AIM EXPRESS short listing	1644
6	Conformance requirements	2012
6.1	Conformance class for Design_physical_characterization (CC 1)	2013
6.2	Conformance class for Design_characterization (CC 2)	2013
6.3	Conformance class for Spatial_arrangement (CC 3)	2013
6.4	Conformance class for Design_characterization_and_spatial_arrangement (CC 4)	2013
6.5	Conformance class for Spatial_arrangement_and_configuration_data (CC 5)	2013
6.6	Conformance class for Documented_design_characterization (CC 6)	2014
6.7	Conformance class for Documented_item_characterization (CC 7)	2014
6.8	Conformance class for Complete_system_characterization (CC 8)	2014
6.9	Conformance class for Product_structure_and_work_management_data (CC 9)	2014
6.10	Conformance class for Design_structure_and_work_mnagement_data (CC 10)	2014
6.11	Conformance class for Location_information_and_work_management_data (CC 11)	2014
6.12	Conformance class for Design_structure_and_configuration_data (CC 12)	2014
6.13	Conformance class for Item_structure_and_configuration_data (CC 13)	2014
6.14	Conformance class for Documented_design_structure_and_configuration_data (CC 14)	2014
6.15	Conformance class for Documented_item_structure_and_configuration_data (CC 15)	2015
Annex A (normative)	AIM EXPRESS expanded listing	2097
Annex B (normative)	AIM short names	2385

Annex C (normative) Implementation method specific requirements .....	2412
Annex D (normative) Protocol Implementation Conformance Statement (PICS) proforma .....	2413
Annex E (normative) Information object registration .....	2416
E.1 Document identification .....	2416
E.2 Schema identification .....	2416
Annex F (informative) Application activity model .....	2417
F.1 Application activity model definitions and abbreviations .....	2417
F.2 Application activity model diagrams .....	2426
Annex G (informative) Application reference model .....	2439
Annex H (informative) AIM EXPRESS-G .....	2533
Annex J (informative) Computer interpretable listings .....	2630
 Bibliography .....	 2631
 Index .....	 2632

## Figures

Figure 1 - Electrical equipment in a tramcar .....	xiv
Figure 2 - Product data that characterizes industrial electrotechnical equipment .....	xvi
Figure 3 - Data planning model .....	xviii
Figure 4 - Functionality of a gas - insulated switchgear .....	22
Figure 5 - Example of an installation; installation of devices on cabletrays or in cabinets .....	23
Figure 6 - Example of a product description; outcut of a wiring harness .....	26
Figure 7 - Hierarchical composition .....	64
Figure 8 - Configuration inheritance scheme .....	91
Figure 9 - Connection_bundle of Connection objects .....	94
Figure 10 - Predefined corner styles .....	102
Figure 11 - Predefined curve end styles .....	103
Figure 12 - Predefined dimension symbols .....	133
Figure 13 - Power transformer fed through an overhead cable .....	186
Figure 14 - Predefined geometrical tolerance symbols .....	205
Figure 15 - Internally wired connector .....	219
Figure 16 - Predefined point marker symbols .....	280

## ISO 10303-212:2001(E)

Figure 17 - Predefined section end shapes.....	333
Figure 18 - Predefined terminator symbols.....	375
Figure 19 - Text characteristics.....	379
Figure 20 - Mirror angle.....	381
Figure 21 - Text appearance and rotation of text.....	383
Figure F.1 - AAM diagram of node A-0: Electrotechnical design and installation.....	2426
Figure F.2 - AAM diagram of node A0: Engineer electrotechnical system .....	2427
Figure F.3 - AAM diagram of node A1: Identify system.....	2428
Figure F.4 - AAM diagram of node A2: Develop conceptual design.....	2430
Figure F.5 - AAM diagram of node A3: Develop system.....	2431
Figure F.6 - AAM diagram of node A4: Implement system .....	2432
Figure F.7 - AAM diagram of node A5: Put system into operation .....	2433
Figure G.1 - ARM diagram in EXPRESS-G: 1 of 93 .....	2440
Figure G.2 - ARM diagram in EXPRESS-G: 2 of 93 .....	2441
Figure G.3 - ARM diagram in EXPRESS-G: 3 of 93 .....	2442
Figure G.4 - ARM diagram in EXPRESS-G: 4 of 93 .....	2443
Figure G.5 - ARM diagram in EXPRESS-G: 5 of 93 .....	2444
Figure G.6 - ARM diagram in EXPRESS-G: 6 of 93 .....	2445
Figure G.7 - ARM diagram in EXPRESS-G: 7 of 93 .....	2446
Figure G.8 - ARM diagram in EXPRESS-G: 8 of 93 .....	2447
Figure G.9 - ARM diagram in EXPRESS-G: 9 of 93 .....	2448
Figure G.10 - ARM diagram in EXPRESS-G: 10 of 93 .....	2449
Figure G.11 - ARM diagram in EXPRESS-G: 11 of 93 .....	2450
Figure G.12 - ARM diagram in EXPRESS-G: 12 of 93 .....	2451
Figure G.13 - ARM diagram in EXPRESS-G: 13 of 93 .....	2452
Figure G.14 - ARM diagram in EXPRESS-G: 14 of 93 .....	2453
Figure G.15 - ARM diagram in EXPRESS-G: 15 of 93 .....	2454
Figure G.16 - ARM diagram in EXPRESS-G: 16 of 93 .....	2455
Figure G.17 - ARM diagram in EXPRESS-G: 17 of 93 .....	2456
Figure G.18 - ARM diagram in EXPRESS-G: 18 of 93 .....	2457
Figure G.19 - ARM diagram in EXPRESS-G: 19 of 93 .....	2458
Figure G.20 - ARM diagram in EXPRESS-G: 20 of 93 .....	2459
Figure G.21 - ARM diagram in EXPRESS-G: 21 of 93 .....	2460
Figure G.22 - ARM diagram in EXPRESS-G: 22 of 93 .....	2461
Figure G.23 - ARM diagram in EXPRESS-G: 23 of 93 .....	2462

Figure G.24 - ARM diagram in EXPRESS-G: 24 of 93 .....	2463
Figure G.25 - ARM diagram in EXPRESS-G: 25 of 93 .....	2464
Figure G.26 - ARM diagram in EXPRESS-G: 26 of 93 .....	2465
Figure G.27 - ARM diagram in EXPRESS-G: 27 of 93 .....	2466
Figure G.28 - ARM diagram in EXPRESS-G: 28 of 93 .....	2467
Figure G.29 - ARM diagram in EXPRESS-G: 29 of 93 .....	2468
Figure G.30 - ARM diagram in EXPRESS-G: 30 of 93 .....	2469
Figure G.31 - ARM diagram in EXPRESS-G: 31 of 93 .....	2470
Figure G.32 - ARM diagram in EXPRESS-G: 32 of 93 .....	2471
Figure G.33 - ARM diagram in EXPRESS-G: 33 of 93 .....	2472
Figure G.34 - ARM diagram in EXPRESS-G: 34 of 93 .....	2473
Figure G.35 - ARM diagram in EXPRESS-G: 35 of 93 .....	2474
Figure G.36 - ARM diagram in EXPRESS-G: 36 of 93 .....	2475
Figure G.37 - ARM diagram in EXPRESS-G: 37 of 93 .....	2476
Figure G.38 - ARM diagram in EXPRESS-G: 38 of 93 .....	2477
Figure G.39 - ARM diagram in EXPRESS-G: 39 of 93 .....	2478
Figure G.40 - ARM diagram in EXPRESS-G: 40 of 93 .....	2479
Figure G.41 - ARM diagram in EXPRESS-G: 41 of 93 .....	2480
Figure G.42 - ARM diagram in EXPRESS-G: 42 of 93 .....	2481
Figure G.43 - ARM diagram in EXPRESS-G: 43 of 93 .....	2482
Figure G.44 - ARM diagram in EXPRESS-G: 44 of 93 .....	2483
Figure G.45 - ARM diagram in EXPRESS-G: 45 of 93 .....	2484
Figure G.46 - ARM diagram in EXPRESS-G: 46 of 93 .....	2485
Figure G.47 - ARM diagram in EXPRESS-G: 47 of 93 .....	2486
Figure G.48 - ARM diagram in EXPRESS-G: 48 of 93 .....	2487
Figure G.49 - ARM diagram in EXPRESS-G: 49 of 93 .....	2488
Figure G.50 - ARM diagram in EXPRESS-G: 50 of 93 .....	2489
Figure G.51 - ARM diagram in EXPRESS-G: 51 of 93 .....	2490
Figure G.52 - ARM diagram in EXPRESS-G: 52 of 93 .....	2491
Figure G.53 - ARM diagram in EXPRESS-G: 53 of 93 .....	2492
Figure G.54 - ARM diagram in EXPRESS-G: 54 of 93 .....	2493
Figure G.55 - ARM diagram in EXPRESS-G: 55 of 93 .....	2494
Figure G.56 - ARM diagram in EXPRESS-G: 56 of 93 .....	2495
Figure G.57 - ARM diagram in EXPRESS-G: 57 of 93 .....	2496
Figure G.58 - ARM diagram in EXPRESS-G: 58 of 93 .....	2497

Figure G.59 - ARM diagram in EXPRESS-G: 59 of 93 .....	2498
Figure G.60 - ARM diagram in EXPRESS-G: 60 of 93 .....	2499
Figure G.61 - ARM diagram in EXPRESS-G: 61 of 93 .....	2500
Figure G.62 - ARM diagram in EXPRESS-G: 62 of 93 .....	2501
Figure G.63 - ARM diagram in EXPRESS-G: 63 of 93 .....	2502
Figure G.64 - ARM diagram in EXPRESS-G: 64 of 93 .....	2503
Figure G.65 - ARM diagram in EXPRESS-G: 65 of 93 .....	2504
Figure G.66 - ARM diagram in EXPRESS-G: 66 of 93 .....	2505
Figure G.67 - ARM diagram in EXPRESS-G: 67 of 93 .....	2506
Figure G.68 - ARM diagram in EXPRESS-G: 68 of 93 .....	2507
Figure G.69 - ARM diagram in EXPRESS-G: 69 of 93 .....	2508
Figure G.70 - ARM diagram in EXPRESS-G: 70 of 93 .....	2509
Figure G.71 - ARM diagram in EXPRESS-G: 71 of 93 .....	2510
Figure G.72 - ARM diagram in EXPRESS-G: 72 of 93 .....	2511
Figure G.73 - ARM diagram in EXPRESS-G: 73 of 93 .....	2512
Figure G.74 - ARM diagram in EXPRESS-G: 74 of 93 .....	2513
Figure G.75 - ARM diagram in EXPRESS-G: 75 of 93 .....	2514
Figure G.76 - ARM diagram in EXPRESS-G: 76 of 93 .....	2515
Figure G.77 - ARM diagram in EXPRESS-G: 77 of 93 .....	2516
Figure G.78 - ARM diagram in EXPRESS-G: 78 of 93 .....	2517
Figure G.79 - ARM diagram in EXPRESS-G: 79 of 93 .....	2518
Figure G.80 - ARM diagram in EXPRESS-G: 80 of 93 .....	2519
Figure G.81 - ARM diagram in EXPRESS-G: 81 of 93 .....	2520
Figure G.82 - ARM diagram in EXPRESS-G: 82 of 93 .....	2521
Figure G.83 - ARM diagram in EXPRESS-G: 83 of 93 .....	2522
Figure G.84 - ARM diagram in EXPRESS-G: 84 of 93 .....	2523
Figure G.85 - ARM diagram in EXPRESS-G: 85 of 93 .....	2524
Figure G.86 - ARM diagram in EXPRESS-G: 86 of 93 .....	2525
Figure G.87 - ARM diagram in EXPRESS-G: 87 of 93 .....	2526
Figure G.88 - ARM diagram in EXPRESS-G: 88 of 93 .....	2527
Figure G.89 - ARM diagram in EXPRESS-G: 89 of 93 .....	2528
Figure G.90 - ARM diagram in EXPRESS-G: 90 of 93 .....	2529
Figure G.91 - ARM diagram in EXPRESS-G: 91 of 93 .....	2530
Figure G.92 - ARM diagram in EXPRESS-G: 92 of 93 .....	2531
Figure G.93 - ARM diagram in EXPRESS-G: 93 of 93 .....	2532



Figure H.1 - AIM diagram in EXPRESS-G: 1 of 96.....	2534
Figure H.2 - AIM diagram in EXPRESS-G: 2 of 96.....	2535
Figure H.3 - AIM diagram in EXPRESS-G: 3 of 96.....	2536
Figure H.4 - AIM diagram in EXPRESS-G: 4 of 96.....	2537
Figure H.5 - AIM diagram in EXPRESS-G: 5 of 96.....	2538
Figure H.6 - AIM diagram in EXPRESS-G: 6 of 96.....	2539
Figure H.7 - AIM diagram in EXPRESS-G: 7 of 96.....	2540
Figure H.8 - AIM diagram in EXPRESS-G: 8 of 96.....	2541
Figure H.9 - AIM diagram in EXPRESS-G: 9 of 96.....	2542
Figure H.10 - AIM diagram in EXPRESS-G: 10 of 96.....	2543
Figure H.11 - AIM diagram in EXPRESS-G: 11 of 96.....	2544
Figure H.12 - AIM diagram in EXPRESS-G: 12 of 96.....	2545
Figure H.13 - AIM diagram in EXPRESS-G: 13 of 96.....	2546
Figure H.14 - AIM diagram in EXPRESS-G: 14 of 96.....	2547
Figure H.15 - AIM diagram in EXPRESS-G: 15 of 96.....	2548
Figure H.16 - AIM diagram in EXPRESS-G: 16 of 96.....	2549
Figure H.17 - AIM diagram in EXPRESS-G: 17 of 96.....	2550
Figure H.18 - AIM diagram in EXPRESS-G: 18 of 96.....	2551
Figure H.19 - AIM diagram in EXPRESS-G: 19 of 96.....	2552
Figure H.20 - AIM diagram in EXPRESS-G: 20 of 96.....	2553
Figure H.21 - AIM diagram in EXPRESS-G: 21 of 96.....	2554
Figure H.22 - AIM diagram in EXPRESS-G: 22 of 96.....	2555
Figure H.23 - AIM diagram in EXPRESS-G: 23 of 96.....	2556
Figure H.24 - AIM diagram in EXPRESS-G: 24 of 96.....	2557
Figure H.25 - AIM diagram in EXPRESS-G: 25 of 96.....	2558
Figure H.26 - AIM diagram in EXPRESS-G: 26 of 96.....	2559
Figure H.27 - AIM diagram in EXPRESS-G: 27 of 96.....	2560
Figure H.28 - AIM diagram in EXPRESS-G: 28 of 96.....	2561
Figure H.29 - AIM diagram in EXPRESS-G: 29 of 96.....	2562
Figure H.30 - AIM diagram in EXPRESS-G: 30 of 96.....	2563
Figure H.31 - AIM diagram in EXPRESS-G: 31 of 96.....	2564
Figure H.32 - AIM diagram in EXPRESS-G: 32 of 96.....	2565
Figure H.33 - AIM diagram in EXPRESS-G: 33 of 96.....	2566
Figure H.34 - AIM diagram in EXPRESS-G: 34 of 96.....	2567
Figure H.35 - AIM diagram in EXPRESS-G: 35 of 96.....	2568

Figure H.36 - AIM diagram in EXPRESS-G: 36 of 96.....	2569
Figure H.37 - AIM diagram in EXPRESS-G: 37 of 96.....	2570
Figure H.38 - AIM diagram in EXPRESS-G: 38 of 96.....	2571
Figure H.39 - AIM diagram in EXPRESS-G: 39 of 96.....	2572
Figure H.40 - AIM diagram in EXPRESS-G: 40 of 96.....	2573
Figure H.41 - AIM diagram in EXPRESS-G: 41 of 96.....	2574
Figure H.42 - AIM diagram in EXPRESS-G: 42 of 96.....	2575
Figure H.43 - AIM diagram in EXPRESS-G: 43 of 96.....	2576
Figure H.44 - AIM diagram in EXPRESS-G: 44 of 96.....	2577
Figure H.45 - AIM diagram in EXPRESS-G: 45 of 96.....	2578
Figure H.46 - AIM diagram in EXPRESS-G: 46 of 96.....	2579
Figure H.47 - AIM diagram in EXPRESS-G: 47 of 96.....	2580
Figure H.48 - AIM diagram in EXPRESS-G: 48 of 96.....	2581
Figure H.49 - AIM diagram in EXPRESS-G: 49 of 96.....	2582
Figure H.50 - AIM diagram in EXPRESS-G: 50 of 96.....	2583
Figure H.51 - AIM diagram in EXPRESS-G: 51 of 96.....	2584
Figure H.52 - AIM diagram in EXPRESS-G: 52 of 96.....	2585
Figure H.53 - AIM diagram in EXPRESS-G: 53 of 96.....	2586
Figure H.54 - AIM diagram in EXPRESS-G: 54 of 96.....	2587
Figure H.55 - AIM diagram in EXPRESS-G: 55 of 96.....	2588
Figure H.56 - AIM diagram in EXPRESS-G: 56 of 96.....	2589
Figure H.57 - AIM diagram in EXPRESS-G: 57 of 96.....	2590
Figure H.58 - AIM diagram in EXPRESS-G: 58 of 96.....	2591
Figure H.59 - AIM diagram in EXPRESS-G: 59 of 96.....	2592
Figure H.60 - AIM diagram in EXPRESS-G: 60 of 96.....	2593
Figure H.61 - AIM diagram in EXPRESS-G: 61 of 96.....	2594
Figure H.62 - AIM diagram in EXPRESS-G: 62 of 96.....	2595
Figure H.63 - AIM diagram in EXPRESS-G: 63 of 96.....	2596
Figure H.64 - AIM diagram in EXPRESS-G: 64 of 96.....	2597
Figure H.65 - AIM diagram in EXPRESS-G: 65 of 96.....	2598
Figure H.66 - AIM diagram in EXPRESS-G: 66 of 96.....	2599
Figure H.67 - AIM diagram in EXPRESS-G: 67 of 96.....	2600
Figure H.68 - AIM diagram in EXPRESS-G: 68 of 96.....	2601
Figure H.69 - AIM diagram in EXPRESS-G: 69 of 96.....	2602
Figure H.70 - AIM diagram in EXPRESS-G: 70 of 96.....	2603

Figure H.71 - AIM diagram in EXPRESS-G: 71 of 96.....	2604
Figure H.72 - AIM diagram in EXPRESS-G: 72 of 96.....	2605
Figure H.73 - AIM diagram in EXPRESS-G: 73 of 96.....	2606
Figure H.74 - AIM diagram in EXPRESS-G: 74 of 96.....	2607
Figure H.75 - AIM diagram in EXPRESS-G: 75 of 96.....	2608
Figure H.76 - AIM diagram in EXPRESS-G: 76 of 96.....	2609
Figure H.77 - AIM diagram in EXPRESS-G: 77 of 96.....	2610
Figure H.78 - AIM diagram in EXPRESS-G: 78 of 96.....	2611
Figure H.79 - AIM diagram in EXPRESS-G: 79 of 96.....	2612
Figure H.80 - AIM diagram in EXPRESS-G: 80 of 96.....	2613
Figure H.81 - AIM diagram in EXPRESS-G: 81 of 96.....	2614
Figure H.82 - AIM diagram in EXPRESS-G: 82 of 96.....	2615
Figure H.83 - AIM diagram in EXPRESS-G: 83 of 96.....	2616
Figure H.84 - AIM diagram in EXPRESS-G: 84 of 96.....	2617
Figure H.85 - AIM diagram in EXPRESS-G: 85 of 96.....	2618
Figure H.86 - AIM diagram in EXPRESS-G: 86 of 96.....	2619
Figure H.87 - AIM diagram in EXPRESS-G: 87 of 96.....	2620
Figure H.88 - AIM diagram in EXPRESS-G: 88 of 96.....	2621
Figure H.89 - AIM diagram in EXPRESS-G: 89 of 96.....	2622
Figure H.90 - AIM diagram in EXPRESS-G: 90 of 96.....	2623
Figure H.91 - AIM diagram in EXPRESS-G: 91 of 96.....	2624
Figure H.92 - AIM diagram in EXPRESS-G: 92 of 96.....	2625
Figure H.93 - AIM diagram in EXPRESS-G: 93 of 96.....	2626
Figure H.94 - AIM diagram in EXPRESS-G: 94 of 96.....	2627
Figure H.95 - AIM diagram in EXPRESS-G: 95 of 96.....	2628
Figure H.96 - AIM diagram in EXPRESS-G: 96 of 96.....	2629

## Tables

Table 1 - RGB values for predefined colors .....	286
Table 2 - Line segment and space lengths for predefined curve font lists.....	287
Table 3 - AIC references for mapping tables.....	665
Table 4 - Mapping table allocation UoF .....	666
Table 5 - Mapping table classification UoF.....	672
Table 6 - Mapping table conditions UoF .....	686
Table 7 - Mapping table configuration_management UoF .....	705
Table 8 - Mapping table course UoF .....	746
Table 9 - Mapping table designation UoF .....	756
Table 10 - Mapping table dimensioned_documentation UoF.....	758
Table 11 - Mapping table documentation UoF .....	777
Table 12 - Mapping table effectivity_data UoF.....	850
Table 13 - Mapping table external_reference UoF .....	965
Table 14 - Mapping table function_structure UoF .....	1008
Table 15 - Mapping table functional_connectivity UoF.....	1020
Table 16 - Mapping table installation UoF .....	1027
Table 17 - Mapping table messages UoF.....	1053
Table 18 - Mapping table network_allocation UoF.....	1074
Table 19 - Mapping table organizational_data UoF .....	1079
Table 20 - Mapping table organizational_data UoF .....	1263
Table 21 - Mapping table physical_connectivity UoF.....	1285
Table 22 - Mapping table product_structure UoF.....	1295
Table 23 - Mapping table properties UoF.....	1359
Table 24 - Mapping table remark UoF.....	1443
Table 25 - Mapping table schematic_documentation UoF .....	1509
Table 26 - Mapping table site UoF .....	1546
Table 27 - Mapping table work_management UoF .....	1564
Table 29 - Usage of AIM elements in conformance classes.....	2019
Table B.1 - AIM short names of entities.....	2385
Table F.1 - AAM - data classes.....	2435

## Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

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

Attention is drawn to the possibility that some of the elements of this part of ISO 10303 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 10303-212 was prepared jointly by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*, and IEC/TC 3, *Information structures, documentation and graphical symbols*. The draft was circulated for voting to the national bodies of both ISO and IEC.

The technical committees involved have agreed not to change any part of this part of ISO 10303 without mutual agreement.

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

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

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

[<http://www.nist.gov/sc4/editing/step/titles/>](http://www.nist.gov/sc4/editing/step/titles/)

Annexes A to E form a normative part of this part of ISO 10303. Annexes F to J are for information only.

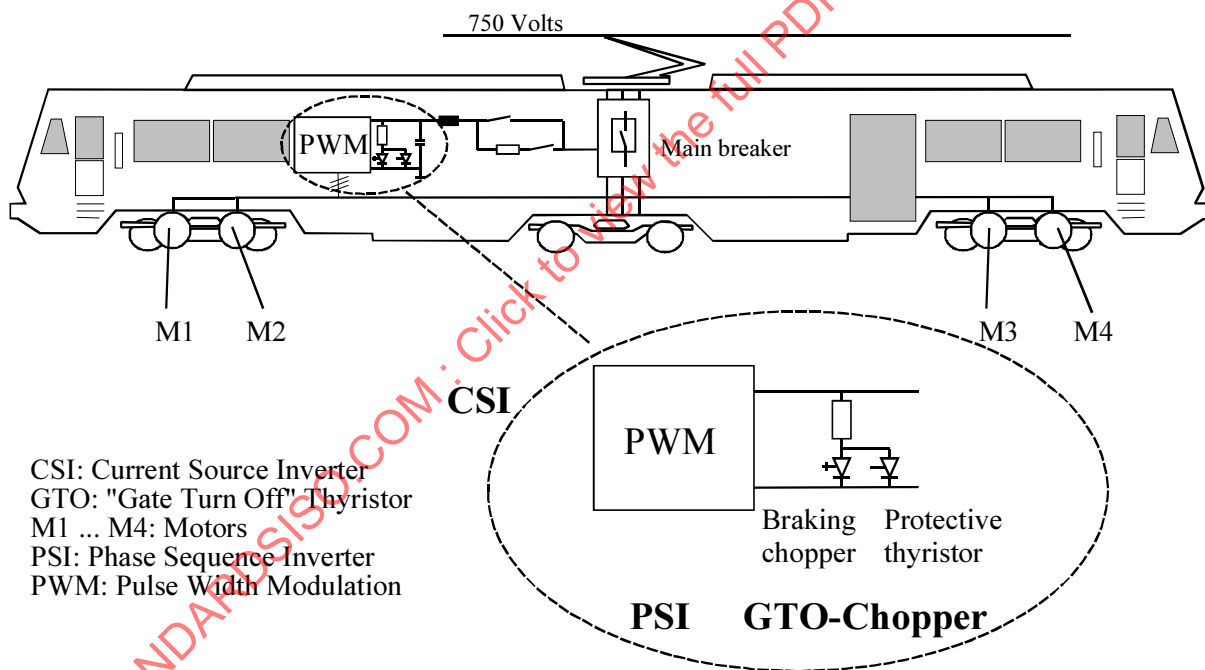
## Introduction

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

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

EXAMPLE 1 The Figure 1 shows electrotechnical equipment in a tramcar.

EXAMPLE 2 In large installations, cables are often customized during the commissioning of an electrotechnical system.



**Figure 1 - Electrical equipment in a tramcar**

This part of ISO 10303 specifies an application protocol (AP) for design and installation information of electrotechnical equipment used in plants, industrial systems or vehicles. This part describes the information shared between the parties involved in the design, the installation and the commissioning of the apparatus. Design is understood as a process of combining components such as relays, programmable logic controllers, or software that comprise a system. Such a system may also be used to control a chemical process in a factory. The description includes various characteristics of the

design, such as functional aspects, physical aspects or the aspects related to the installation of the equipment.

The detailed product information of procured or premanufactured parts is not included in the AP but the information needed to customize products accordingly to the requirements of the installation is included.

**EXAMPLE 3** Figure 2 shows some product data that characterize an electrotechnical system. The circuitry essentially comprises the product data for main circuits and the control circuits together with the circuits for control, indication and monitoring. In Figure 2 the functions describe functional building blocks that specify the logical structure of the system, their connectivity and the signals used. The arrangement data describe how the components that make up the system are physically arranged.

Figure 3 contains the data planning model that provides a high level description of the requirements for this application protocol, as well as the relationships between the basic data components.

The planning model illustrates that an electrotechnical system is described by function-oriented product data and product-oriented product data. Both descriptions may be allocated to each other. Arrangement data, the routing of cables and the arrangement of the equipment is described by the UoFs installation, course, and site. In many cases the system components are categorized in accordance to an existing classification system. The UoF classification specifies the necessary concepts.

The information flow within the system may be described by specifying the signals that are generated, processed or transmitted within the system. The UoF messages comprises the concepts that specify the information flow.

The UoF remark allows to assign explanatory information to the system or its constituents. Organizational data such as approval or manufacturer is described by the UoF organizational\_data.

Requirements and constraints levied against the system are represented by conditions that affect the products used, their functionality and their installation. Configuration\_management, organizational\_data, and work\_management allow the handling of data concerning information about variants and options of the design, releases and approvals, or the appointment of work.

The documentation of the system is addressed by the documentation and dimensioned\_documentation data. Designation data allows the identification of the equipment or its functions uniquely within the system. Technical data may be addressed by assigning property data to the elements of the data model.

This application protocol defines the context, scope, and information requirements for the exchange of design and installation information of electrotechnical equipment and specifies the integrated resources necessary to satisfy these requirements.

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. Clause 3 lists the words defined in this part of ISO 10303 and gives pointers to words defined elsewhere. An application activity model that is the basis for the definition of the scope is provided in annex F. The information requirements of the application are specified in clause

4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex G.

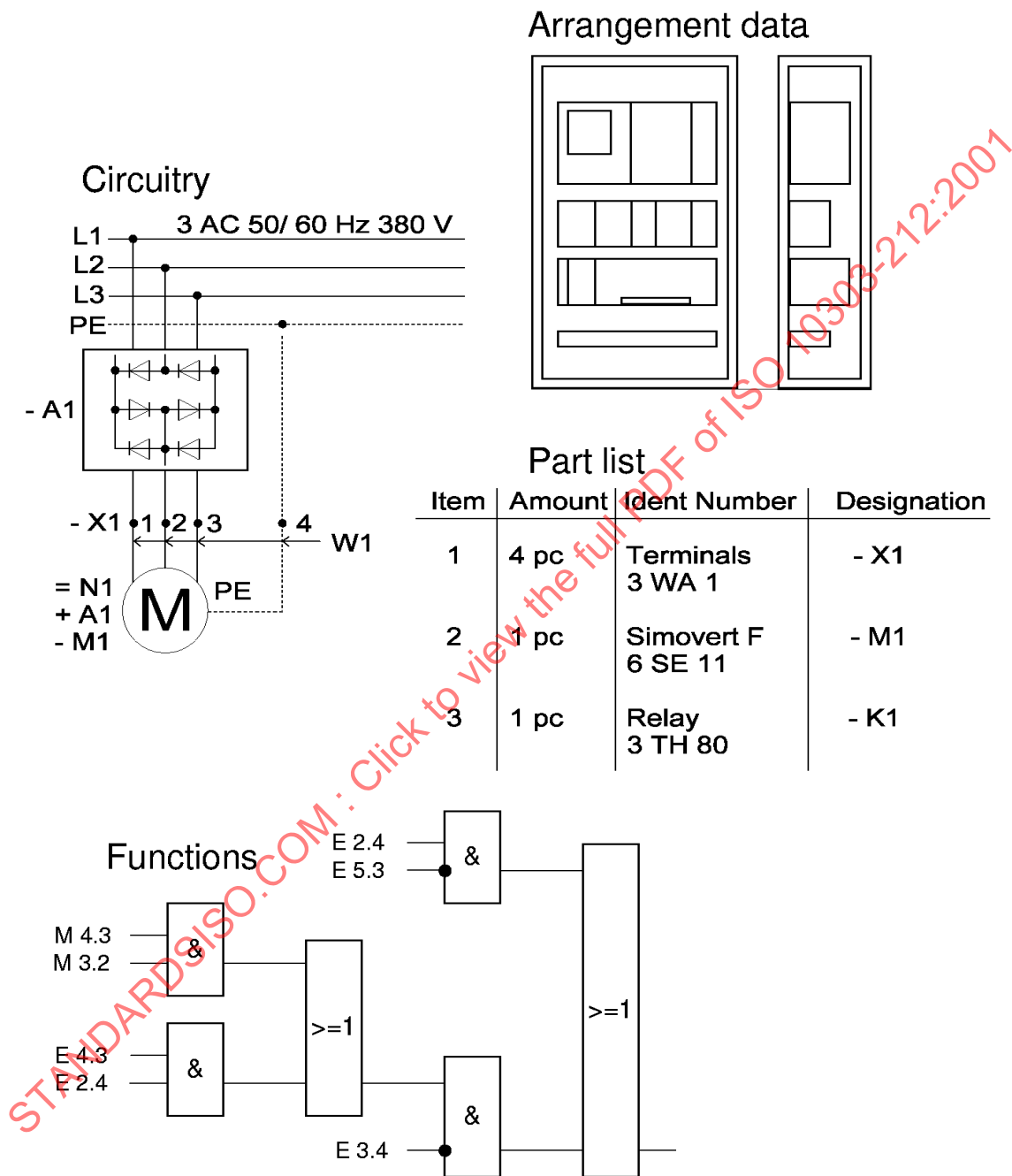


Figure 2 - Product data that characterizes industrial electrotechnical equipment



Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in 5.1, shows the correspondence between the information requirements and the AIM. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. The expanded listing given in Annex A contains the complete EXPRESS for the AIM without annotation. A graphical representation of the AIM is given in annex H. Additional requirements for specific implementation methods are given in annex C.

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-212:2001

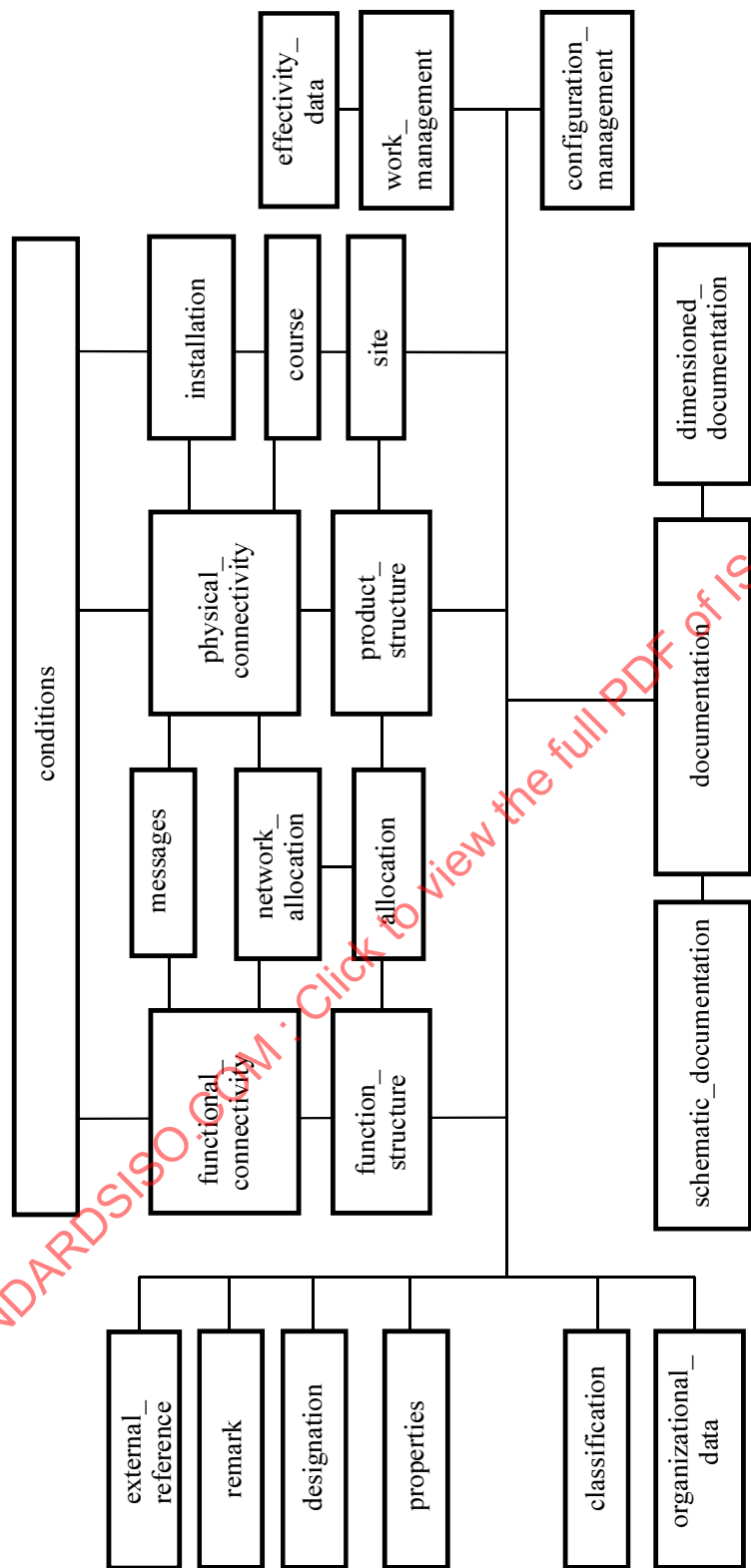


Figure 3 - Data planning model

# Industrial automation systems and integration — Product data representation and exchange — Part 212: Application protocol: Electrotechnical design and installation

## 1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for the design, installation, and commissioning of electrotechnical systems. This Application protocol does not impose any restriction on the usage of these systems. Equipment for power-transmission, -distribution, and -generation, electrical machinery, electric light, electric heat, control and automation systems is in scope.

EXAMPLE 1 Electrotechnical systems may be used in plants, buildings, or transportation systems such as cars or ships, etc.

The information content of documents that describe a system in accordance to IEC 61082: *Preparation of documents used in electrotechnology* is part of the scope. Also included is the decomposition of products and functions, their interrelations, their connectivity and their schematic representation.

NOTE 1 The application activity model in annex F provides a graphical representation of the processes and information flows that are the basis for the definition of the scope of this part of ISO 10303.

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

— Data to describe an electrotechnical system throughout design, commissioning and delivery phases;

NOTE 2 Throughout the commissioning of a system an operating time of significant duration may occur. Such a stage is often requested by the customer to verify the operating performance of the system. The data required to keep track of design changes or maintenance activities are considered to be part of the data that describe commissioning of the system.

— Data to specify the equipment used in the electrotechnical system;

— Data to describe terminals and interfaces of the equipment;

— Data to describe the functional decomposition of the system;

— Data to describe the connectivity and cabling of devices and equipment;

- Data to specify the installation of the equipment, including physical locations and information about the shape of the equipment;
- Data for the reference designation of the design's functional modules and of the equipment used;
- Data to specify the pieces of information exchanged between the various constituents of the design;
- Data to provide the design with actual and planned technical characteristics;
- Data to document the system or its design and to keep track of version and status of the documentation;
- Data for the tracking of a design's release;
- Data to track the approval of functional and physical objects used in the system;
- Data to perform the configuration management of the equipment;
- Data to describe the requirements levied upon the design and their allocation to functional objects, physical objects, and the physical implementation;
- Data to provide information about the contractual basis and the work packages assigned to develop and implement the design;
- Data to classify and categorize the functional and physical objects that make up the electrotechnical system in accordance to standardized or user specific classification systems;
- Data to the assignment of comments or instructions to the product data;

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

- Detailed product information or manufacturing information of procured premanufactured parts;
- Data for the simulation and testing of electrotechnical systems;

EXAMPLE 2 Information about test patterns, behavioural models, etc.

- Detailed mechanical design information of electric/electronic products;
- The management of the process used for the design of electrotechnical systems;
- The process plans for the procurement, assembly or shipping of electrotechnical systems or their constituents;
- The administrative procurement and cost data used by an enterprise.

## 2 Normative references

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

IEC 61082-1:1991, *Preparation of documents used in electrotechnology — Part 1: General requirements*.

IEC 61360:1995, *Standard data element types with associated classification scheme for electric components*.

ISO 639:1988, *Code for the representation of names of languages*.

ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange*.

ISO/IEC 6937:1994, *Information technology — Coded character set for text communication — Latin alphabet*.

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

ISO/IEC 8859-1:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*.

ISO/IEC 10646, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*.

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

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

ISO 10303-21:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*.

ISO 10303-31:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts*.

## ISO 10303-212:2001(E)

ISO 10303-32:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 32: Conformance testing methodology and framework: Requirements on testing laboratories and clients.*

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

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

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

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

ISO 10303-45:1998, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resource: Materials.*

ISO 10303-46:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 46: Integrated generic resources: Visual presentation.*

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

ISO 10303-504:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 504: Application interpreted construct: Draughting annotation.*

ISO 10303-506:2000, *Industrial automation systems and integration — Product data representation and exchange — Part 506: Application interpreted construct: Draughting elements.*

### 3 Terms, definitions, and abbreviations

For the purposes of this International Standard, the following definitions apply.

#### 3.1 Terms defined in IEC 61082-1

For the purposes of this part of ISO 10303, the following terms defined in IEC 61082-1 apply.

- attached representation;
- block diagram;
- circuit diagram;
- connection diagram;
- detached representation;
- diagram;
- document;
- function diagram;
- overview diagram.

#### 3.2 Terms defined in ISO 10303-1

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

- application;
- application activity model (AAM);
- application interpreted model (AIM);
- application object;
- application protocol (AP);
- application reference model (ARM);
- conformance testing;
- implementation method;
- integrated resource;
- model;
- PICS proforma;
- product;

## **ISO 10303-212:2001(E)**

- product data;
- unit of functionality (UoF).

### **3.3 Terms defined in ISO 10303-31**

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-31 apply.

- conformance class.

### **3.4 Terms defined in ISO 10303-42**

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-42 apply.

- b spline curve;
- composite curve;
- conic;
- coordinate space;
- curve;
- line;
- polyline;
- trimmed curve.

### **3.5 Terms defined in ISO 10303-44**

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-44 apply.

- constituent.

### **3.6 Terms defined in ISO 10303-46**

For the purposes of this part of ISO 10303, the following terms defined in ISO 10303-46 apply.

- layer;
- presentation;
- visualization.

### **3.7 Other terms and definitions**

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



### 3.7.1

#### **circuit**

an arrangement of networks, devices, or media that allows the flow or association of information, energy, or material within a physical or non-physical product

### 3.7.2

#### **coded character set**

a set of unambiguous rules that establishes a character set and the one-to-one relationship between each character of the set and its coded representation

### 3.7.3

#### **component**

an occurrence of a product used in an assembled product

NOTE A component may be a piece of equipment or a functional module.

### 3.7.4

#### **connectivity**

the entire arrangement of conductors or networks for joining ports, terminals or other conductors

### 3.7.5

#### **electrotechnical equipment**

the apparatus that supports the distribution, generation, storage, transformation, or utilization of electrical energy

### 3.7.6

#### **electrotechnical system**

a system performing its objectives by using mainly electrotechnical equipment

### 3.7.7

#### **function block**

a base element into which the functionality of an electrotechnical system may be decomposed

NOTE A function block may be decomposed into other function blocks.

### 3.7.8

#### **functionality**

the intended purpose or use of a piece of equipment

### 3.7.9

#### **human-interpretable string**

a sequence of alphanumeric characters intended to be interpreted by humans only, even though the information may be stored or displayed by a computer

### 3.7.10

#### **GIS position**

a position specified through a global information system

STANDARDSISO.COM : Click to view the full PDF of ISO 10303-212:2001

### 3.7.11

#### **industrial system**

a system designed or suitable for industrial use

EXAMPLE Systems designed to be used in a production environment with a high resistance against shock or magnetic fields.

### 3.7.12

#### **interface**

a complex access mechanism to the functionality of a component

NOTE In most cases an interface comprises a combination of one or more ports.

### 3.7.13

#### **occurrence**

the use of a typical item at a specific place in a design. Each occurrence is a separate item that refers to the typical item

EXAMPLE Potentiometers is a data class containing catalog items. One instance of the class of potentiometers specifies a potentiometer of type P104 with a resistance of 400 Ohm. The data class of occurrences contains two instances specifying potentiometers R1 and R2 of type P104 that are currently in use. R1 is set to a potential divider ratio of 75% and R2 is set to a potential divider ratio of 10%. Both have the resistance of 400 Ohm because both are occurrences of P104.

### 3.7.14

#### **port**

an access point to a functionality

NOTE Even though, in most cases, a port is implemented by the use of one or more terminals, it is not required for a port to have a physical equivalent.

### 3.7.15

#### **system**

a set of interdependent elements constituted to achieve a given objective by performing a specified function. This includes the equipment used, as well as non-physical items like software, that ensures proper control and interaction of the equipment

### 3.7.16

#### **terminal**

an access point to a piece of equipment

### 3.8 Abbreviations

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

- AAM application activity model;
- AIM application interpreted model;
- AP application protocol;
- ARM application reference model;
- ASCII American Standard Code for Information Interchange
- CAD computer aided design;
- GIS geographic information system;
- id identifier;
- RGB a colorimetric system identifying colours by specifying their red, green and blue portions;
- UoF unit of functionality.

## 4 Information requirements

This subclause specifies the information required for the design, installation, and commissioning of electrotechnical systems.

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTE 1 A graphical representation of the information requirements is given in annex G.

NOTE 2 The information requirements correspond to those of the activities identified as being within the scope of this application protocol in annex F.

NOTE 3 The mapping table specified in 5.1 shows how the integrated resources and application interpreted constructs are used to meet the information requirements of this application protocol.

### 4.1 Units of functionality

This subclause specifies the units of functionality (UoF) for the Electrotechnical design and installation application protocol. This part of ISO 10303 specifies the following units of functionality:

- allocation (AL1);
- classification (CA1);
- conditions (CD1);
- configuration\_management (CF1);
- course (CO1);
- designation (DE1);
- dimensioned\_documentation (DI1);
- documentation (DO1);
- effectivity\_data (EF1);
- external\_reference (ER1);
- function\_structure (F1);
- functional\_connectivity (C1);
- installation (IN1);
- messages (M1);

## ISO 10303-212:2001(E)

- network\_allocation (NA1);
- organizational\_data (OD1);
- physical\_connectivity (PC1);
- product\_structure (PD1);
- properties (PR1);
- remark (R1);
- schematic\_documentation (SC1);
- site (SI1);
- work\_management (W1).

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in 4.2.

### 4.1.1 allocation (AL1)

The allocation UoF defines information that supports assignment of the elements designated in both the function\_structure UoF and the product\_structure UoF. The allocation UoF provides the relationship between the description of the equipment and the appropriate functional description of that equipment.

NOTE Allocation of connectivity is not in the scope of this UoF.

The following application objects are used by the allocation UoF:

- Functional\_unit\_allocation;
- Offered\_function\_allocation;
- Preferred\_item\_allocation.

### 4.1.2 classification (CA1)

The classification UoF specifies all concepts to categorize the packages of information contained in the product data. This includes the information about the documents that specify the classification method.

The following application objects are used by the classification UoF:

- Class\_reference;
- Classification\_association;
- Classification\_attribute;
- Classification\_system;

- General\_classification;
- General\_classification\_hierarchy.

### 4.1.3 conditions (CD1)

The conditions UoF describes the information that is specified in requirements levied against the design or its implementation.

NOTE Requirements laid down in textual form, as property values, or as interface requirements are included.

The following application objects are used by the conditions UoF:

- Requirement;
- Requirement\_assignment;
- Requirement\_document\_assignment;
- Requirement\_relationship.

### 4.1.4 configuration\_management (CF1)

The configuration\_management UoF specifies the concepts for the specification of the planned use of equipment and on the configurations the equipment may be used in. Specification (see 4.2.323) of variants are included in the scope of the configuration\_management UoF.

The following application objects are used by the configuration\_management UoF:

- Alternative\_solution;
- Class\_category\_association;
- Class\_condition\_association;
- Class\_inclusion\_association;
- Class\_specification\_association;
- Class\_structure\_relationship;
- Complex\_product;
- Complex\_product\_relationship;
- Component\_placement;
- Configuration;
- Descriptive\_specification;
- Device\_relationship;

## ISO 10303-212:2001(E)

- Final\_solution;
- Instance\_placement;
- Product\_class;
- Product\_class\_relationship;
- Product\_component;
- Product\_identification;
- Product\_specification;
- Product\_structure\_relationship;
- Solution\_instance\_assignment;
- Specification;
- Specification\_category;
- Specification\_category\_hierarchy;
- Specification\_expression;
- Specification\_inclusion;
- Supplier\_solution;
- Technical\_solution.

### 4.1.5 course (CO1)

The course UoF specifies the data used to describe a three-dimensional path to specify the path of pieces of equipment.

EXAMPLE Examples for pieces of equipment with an assigned path are cable ducts or wireways.

The following application objects are used by the course UoF:

- Curve\_3d;
- Node;
- Node\_relationship;
- Path;
- Path\_node;
- Path\_node\_relationship;



- Path\_relationship;
- Path\_segment.

#### 4.1.6 designation (DE1)

The designation UoF describes concepts used to identify equipment items and function blocks. An item designator may be composed from other designators. The concepts to describe this structure and methods to categorize the constituents of an item designation are in the scope of this UoF. Identification of documents, signals or locations is also addressed by the designation UoF.

The following application objects are used by the designation UoF:

- Document\_designation;
- Object\_designation;
- Object\_designation\_relationship;
- Object\_reference\_designation;
- Signal\_designation;
- Terminal\_designation.

#### 4.1.7 dimensioned\_documentation (DH)

The dimensioned\_documentation UoF specifies concepts to present the electrotechnical system or parts thereof in pictorial or textual form. Dimensioned drawings or schematic diagrams may be used for that purpose.

EXAMPLE Examples for drawings addressed by dimensioned\_documentation UoF are installation diagrams, ground plans, etc.

The following application objects are used by the dimensioned\_documentation UoF:

- Angular\_dimension;
- Chained\_dimension\_pair;
- Curve\_dimension;
- Datum\_feature\_callout;
- Datum\_target\_callout;
- Diameter\_dimension;
- Dimension;
- Dimension\_callout;
- Dimension\_line;

## ISO 10303-212:2001(E)

- Dimension\_line\_terminator;
- Dimension\_sequence\_pair;
- Dimension\_symbol;
- Directed\_curve;
- Draughting\_callout;
- Geometrical\_tolerance;
- Geometrical\_tolerance\_symbol;
- Leader;
- Leader\_directed\_dimension;
- Leader\_terminator;
- Linear\_dimension;
- Ordinate\_dimension;
- Parallel\_dimension\_pair;
- Point\_marker\_symbol;
- Predefined\_symbol;
- Projection\_line;
- Radius\_dimension;
- Structured\_dimension\_callout;
- Terminator\_symbol;
- Unstructured\_dimension\_callout.

### 4.1.8 documentation (DO1)

The documentation UoF specifies concepts to present the electrotechnical system or parts thereof in pictorial or textual form using two-dimensional drawings.

NOTE The concepts used to visualize dimensioned drawings are not included in this UoF.

EXAMPLE Examples for drawings addressed by documentation UoF are circuit diagrams, connection lists, terminal diagrams, etc.

The following application objects are used by the documentation UoF:

- Annotation\_curve;

- Annotation\_element;
- Annotation\_placed\_annotation;
- Annotation\_subfigure;
- Annotation\_subfigure\_definition;
- Annotation\_subfigure\_definition\_element;
- Annotation\_symbol;
- Appearance;
- Cartesian\_coordinate\_space\_2d;
- Colour;
- Curve\_2d;
- Curve\_appearance;
- Draughting\_annotation;
- Draughting\_model;
- Drawing;
- Drawing\_assignment;
- Drawing\_sequence;
- Drawing\_sheet;
- Drawing\_sheet\_layout;
- Drawing\_sheet\_relationship;
- Drawing\_view;
- Externally\_defined\_hatching;
- Externally\_defined\_line\_font;
- Externally\_defined\_symbol;
- Externally\_defined\_text\_font;
- Externally\_defined\_tile;
- Externally\_defined\_tiling;
- Fill\_area;

## ISO 10303-212:2001(E)

- Fill\_area\_appearance;
- Fill\_area\_boundary;
- Group;
- Group\_annotation\_element;
- Group\_element;
- Hatching\_pattern;
- Layer;
- Line\_font;
- Model\_placed\_annotation;
- Point\_2d;
- Predefined\_colour;
- Predefined\_line\_font;
- Predefined\_text\_font;
- Rectangular\_area;
- Sheet\_placed\_annotation;
- Solid\_fill\_area;
- Sub\_group;
- Text;
- Text\_appearance;
- Text\_font;
- Text\_string;
- Tile;
- User\_defined\_colour;
- User\_defined\_hatching;
- User\_defined\_line\_font;
- User\_defined\_symbol;
- User\_defined\_symbol\_definition;

- User\_defined\_tile;
- User\_defined\_tiling;
- View\_displayed\_model;
- View\_placed\_annotation;
- Visibility.

#### 4.1.9 effectivity\_data (EF1)

The effectivity\_data UoF provides the capability to represent information concerning the validity of data. The validity of data can be expressed by effectivities that specify time ranges within which data may be used. Retention periods specify how long data have to be kept and when they may be deleted. Both concepts can use explicit dates or dates expressed by events to represent the relevant points in time.

The following application objects are used by the effectivity\_data UoF:

- Dated\_configuration;
- Duration;
- Effectivity;
- Effectivity\_assignment;
- Effectivity\_relationship;
- Event\_reference;
- Lot\_configuration;
- Manufacturing\_configuration;
- Product\_class;
- Product\_identification;
- Product\_design;
- Retention\_period;
- Serial\_configuration.

#### 4.1.10 external\_reference (ER1)

The external\_reference UoF specifies a reference mechanism to assign additional information to the product data. This information may be available in either electronic or nonelectronic form.

## ISO 10303-212:2001(E)

NOTE Content or format of the referenced data need not be in accordance with this part nor with any other part of ISO 10303.

EXAMPLE Simulation data can be assigned to the product data by using the external\_-reference UoF.

The following application objects are used by the external\_reference UoF:

- Coded\_size;
- Digital\_document;
- Digital\_file;
- Document;
- Document\_assignment;
- Document\_content\_property;
- Document\_creation\_property;
- Document\_file;
- Document\_file\_relationship;
- Document\_format\_property;
- Document\_location\_property;
- Document\_size\_property;
- Document\_type\_property;
- Document\_representation;
- Document\_structure;
- Document\_version;
- Document\_version\_relationship;
- External\_file\_id\_and\_location;
- Hardcopy;
- Language;
- Physical\_document;
- Rectangular\_size;
- Specific\_document\_classification.

### 4.1.11 function\_structure (F1)

The function\_structure UoF specifies the concepts for the description of the functional decomposition of an electrotechnical system. The data describes a segment of an electrotechnical system independent of the specific product used in that segment. The functionality that the segment is to perform is identified. The functions may be either primitive, i.e., its internal structure is not further described, or it may be composed from other functions.

NOTE The concepts for the specification of the connectivity among the function blocks are not part of the function\_structure UoF.

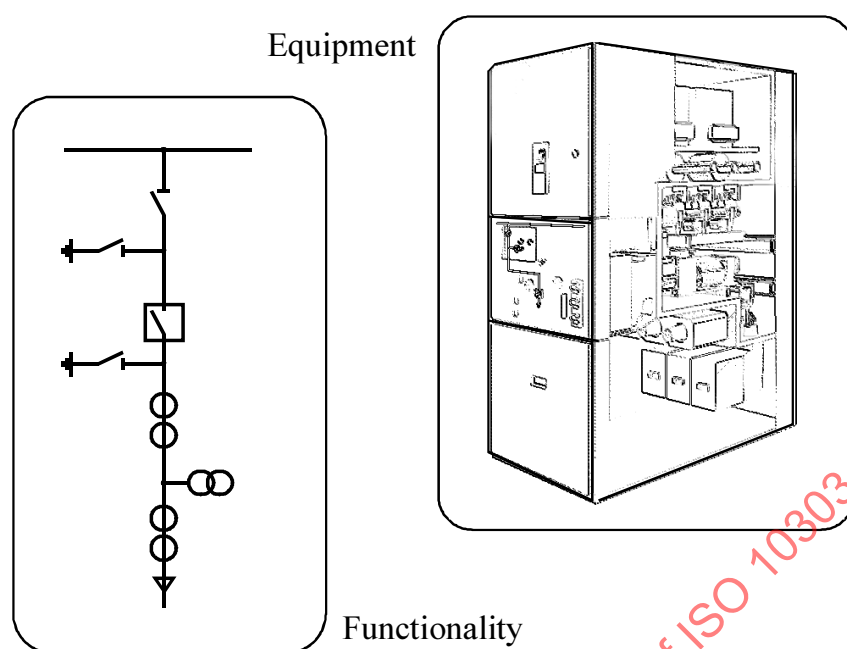
The following application objects are used by the function\_structure UoF:

- Composition\_relationship;
- Function\_definition;
- Function\_definition\_relationship;
- Function\_unit;
- Function\_unit\_relationship;
- Function\_version;
- Function\_version\_relationship;
- Functionality;
- Single\_function\_unit;
- Specified\_function\_unit.

### 4.1.12 functional\_connectivity (FC1)

The functional\_connectivity UoF specifies all those concepts that are required to specify the connectivity among the function blocks describing an electrotechnical system. The specification of the interface through which a function block can be interconnected with other function blocks and the concepts to specify the connectivity among them are included.

EXAMPLE Figure 4 shows the functionality of a gas-insulated switchgear. The functionality is described by the function\_structure UoF and functional\_connectivity UoF. The product data that describes the switchgear comprises concepts from product\_structure UoF and physical\_connectivity UoF.



**Figure 4 - Functionality of a gas - insulated switchgear**

The following application objects are used by the functional\_connectivity UoF:

- Function\_interface;
- Functional\_connectivity\_definition;
- Functional\_connectivity\_definition\_relationship;
- Interface\_port;
- Interface\_port\_relationship;
- Interface\_port\_connectivity;
- Network;
- Port;
- Port\_association;
- Port\_relationship;

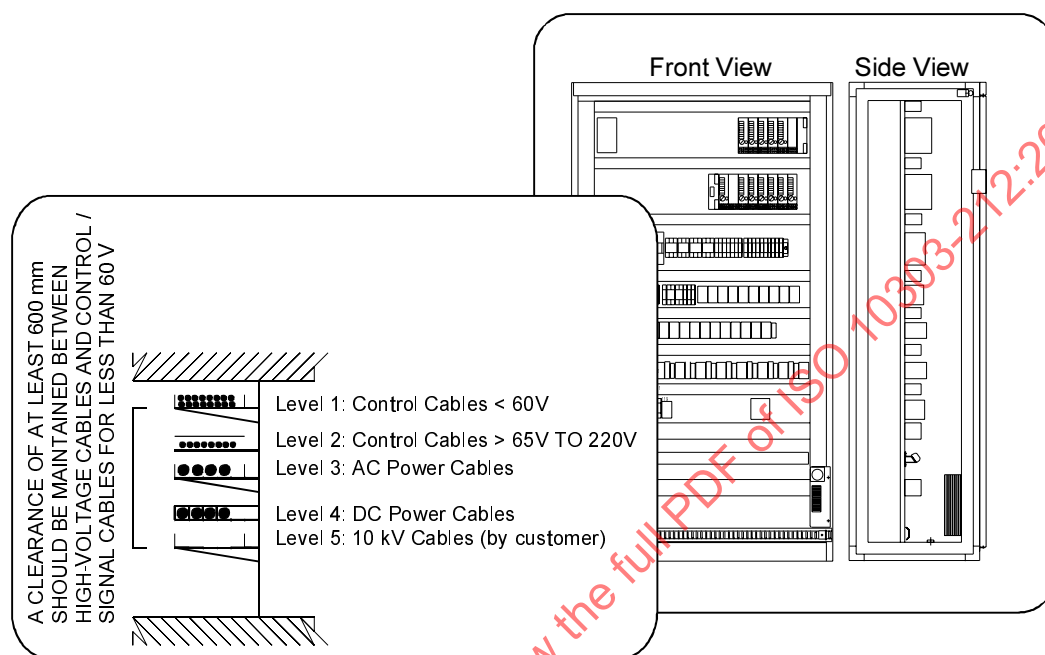
#### **4.1.13 installation (IN1)**

The installation UoF describes the arrangement of the items that make up the electrotechnical system. The data needed for cable routing and cable pulling is in the scope of the installation UoF. Concepts that specify routes are also in the scope of the installation UoF.



EXAMPLE Figure 5 shows sectional views of an electrical installation using cabletrays and control cabinets. The arrangement of the equipment and the laying of the cables is described by using concepts from installation UoF, course UoF, and site UoF.

NOTE The content of the text shown in Figure 5 is for presentation purposes only. Its meaning is not essential for this part of 10303.



**Figure 5 - Example of an installation; installation of devices on cabletrays or in cabinets**

The following application objects are used by the installation UoF:

- Cable\_pull\_information;
- General\_location\_relationship;
- Free\_segment;
- Route;
- Route\_relationship;
- Routed\_object;
- Routed\_segment;
- Section;
- Section\_end;

## ISO 10303-212:2001(E)

- Section\_interface;
- Section\_interface\_relationship;
- Section\_relationship.

### 4.1.14 messages (M1)

The messages UoF specifies the concepts used to describe the flow of information within an electrotechnical system. The concepts to specify the physical representation of the information within the system and the concepts that specify the information content are included. The physical parameters that cause the information flow can be also described.

The following application objects are used by the messages UoF:

- Notification;
- Notification\_relationship;
- Preferred\_equipment\_assignment;
- Process\_variable;
- Process\_variable\_relationship;
- Process\_variable\_system\_assignment;
- Signal;
- Signal\_relationship;
- Signal\_system\_assignment;
- Signal\_value.

### 4.1.15 network\_allocation (NA1)

The network\_allocation UoF defines information that supports the allocation of functional connectivity information to the connectivity established by the equipment used in the electrotechnical system. This includes the data to relate the functional connect nodes to their physical counterparts.

The following application objects are used by the network\_allocation UoF:

- Connectivity\_allocation;
- Port\_allocation;
- Preferred\_item\_terminal\_allocation.

#### 4.1.16 organizational\_data (OD1)

The organizational\_data UoF specifies the concepts to assign organizational information to the product data. This information provides the basis for the management of the product data.

EXAMPLE Examples for organizational information are approval information, information about ownership, time stamps, etc.

The following application objects are used by the organizational\_data UoF:

- Address;
- Approval;
- Approval\_relationship;
- Approval\_status;
- Certification;
- Date\_and\_person\_assignment;
- Date\_and\_person\_or\_organization;
- Date\_time;
- Date\_time\_assignment;
- Date\_time\_interval\_assignment;
- Duration;
- Organization;
- Organization\_relationship;
- Person;
- Person\_in\_organization;
- Person\_in\_organization\_relationship;
- Person\_organization\_assignment;
- Security\_classification;
- Security\_level;
- Time\_interval.

#### 4.1.17 physical\_connectivity (PC1)

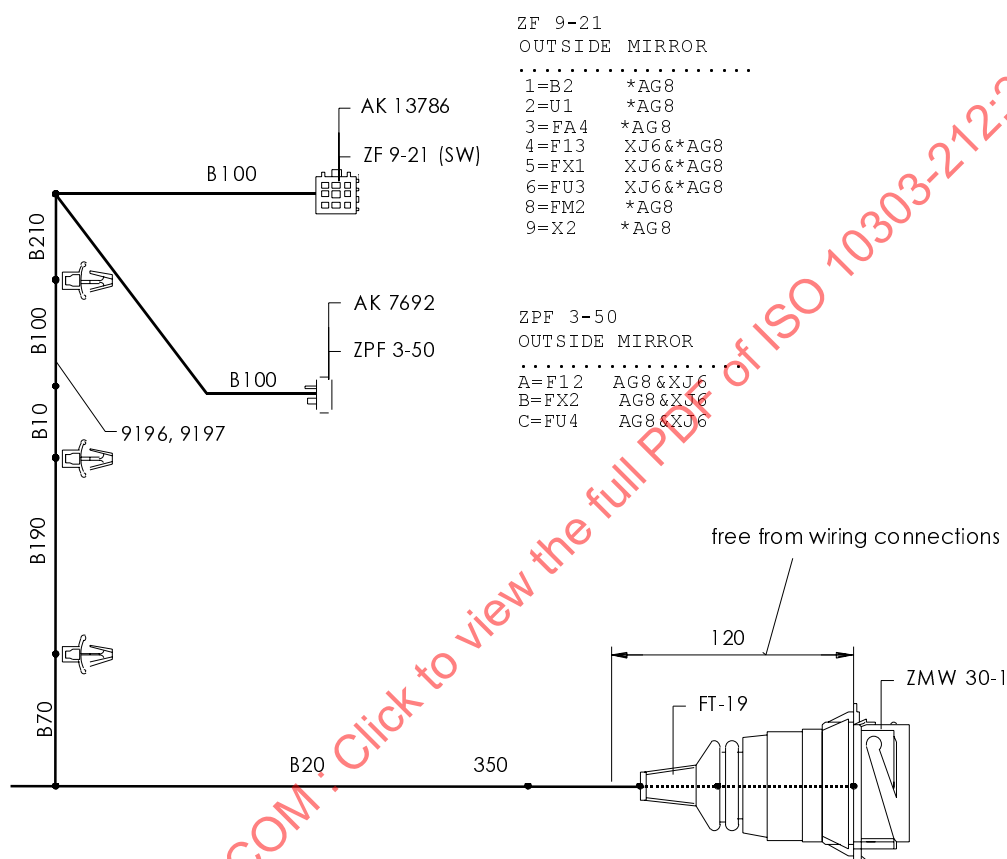
The physical\_connectivity UoF specifies the concepts required to specify the connectivity among the equipment within an electrotechnical system. The specification of the interface through which a piece

## ISO 10303-212:2001(E)

of equipment may be interconnected with other devices and the concepts to specify the connectivity among the equipment are included.

**EXAMPLE** Figure 6 shows a wiring harness of a vehicle. To describe its assembly structure and connectivity concepts of product\_structure UoF and physical\_connectivity UoF are used.

**NOTE** The content of the text shown in Figure 6 is for presentation purposes only. Its meaning is not essential for this part of 10303.



**Figure 6 - Example of a product description: output of a wiring harness**

The following application objects are used by the physical\_connectivity UoF:

- Connection;
- Connectivity\_definition;
- Connectivity\_definition\_relationship;
- Interface;
- Interface\_terminal;
- Interface\_terminal\_relationship;
- Interface\_terminal\_connection;

- Predefined\_connection;
- Terminal;
- Terminal\_relationship.

#### 4.1.18 product\_structure (PD1)

The product\_structure UoF specifies the concepts for the description of the hierarchical structure of the equipment used within an electrotechnical system.

NOTE The concepts for the specification of connectivity among products are not part of the product\_structure UoF.

The following application objects are used by the product\_structure UoF:

- Alias\_designation;
- Alias\_identification;
- Alias\_version;
- Alternate\_item\_relationship;
- Application\_context;
- Assembly\_component\_relationship;
- Assembly\_substitute\_relationship;
- Assembly\_definition;
- Component\_placement;
- Design\_discipline\_item\_definition;
- Device;
- Item;
- Item\_definition\_relationship;
- Item\_identification;
- Item\_version;
- Item\_version\_relationship;
- Make\_from\_relationship;
- Next\_higher\_assembly;
- Numerical\_precision;

## ISO 10303-212:2001(E)

- Part;
- Physical\_assembly\_relationship;
- Physical\_instance;
- Product\_constituent;
- Promissory\_usage;
- Quantified\_device;
- Selected\_device;
- Shape;
- Shape\_assignment;
- Single\_device;
- Specific\_classification\_hierarchy;
- Specific\_item\_classification;
- Specific\_item\_classification\_hierarchy;
- Specified\_device;
- Technical\_system;
- Technical\_system\_relationship.

### 4.1.19 properties (PR1)

The properties UoF specifies a generic concept to assign technical or management data to the information packages that make up the product data.

The following application objects are used by the properties UoF:

- Aggregated\_value;
- Binary\_value;
- Body\_breadth;
- Body\_height;
- Body\_length;
- Component\_colour;
- Cross\_section;

- Data\_element;
- Data\_element\_association;
- Data\_element\_definition;
- Data\_element\_definition\_relationship;
- Data\_element\_relationship;
- Data\_element\_specification;
- Data\_element\_value;
- External\_library\_reference;
- Format\_of\_value;
- Logical\_value;
- Mass;
- Material;
- Mounting\_features;
- Numerical\_value;
- Operating\_temperature;
- Outside\_diameter;
- Predefined\_data\_element;
- Property\_reference;
- Rated\_current;
- Rated\_power;
- Rated\_voltage;
- Single\_value;
- Storage\_temperature;
- String\_value;
- User\_defined\_data\_element.

#### 4.1.20 remark (R1)

The remark UoF specifies the concepts to associate additional human-interpretable information to the product data. Markings applied to the equipment are also supported by the remark UoF.

EXAMPLE Examples for the information supported by remark UoF include information such as comments, assembly instructions, manufacturing instructions, etc.

The following application objects are used by the remark UoF:

- Generic\_note;
- Linear\_pattern\_location;
- Marking;
- Multi\_language\_note;
- Note;
- Note\_association;
- Set\_of\_notes.

#### 4.1.21 schematic\_documentation (SC1)

The schematic\_documentation UoF specifies the concepts used to present product data in schematic diagrams.

NOTE The concepts used to visualize dimensioned drawings are not included in this UoF.

EXAMPLE Examples for schematic constructs are symbols, schematic connections, or cross references.

The following application objects are used by the schematic\_documentation UoF:

- Cartesian\_coordinate\_space\_with\_grid;
- Connect\_area;
- Connecting\_line;
- Cross\_reference;
- Detached\_representation\_reference;
- Direction\_range;
- Item\_presentation;
- Note\_reference;
- Page\_connector;



- Page\_connector\_presentation;
- Page\_connector\_reference;
- Reference\_grid;
- Reference\_grid\_layout;
- Schematic\_node;
- Schematic\_text;
- Typical\_schematic\_node;
- Typical\_schematic\_text.

#### 4.1.22 site (SI1)

The site UoF describes the location where the equipment is positioned. Location (see 4.2.192) is understood as a defined volume of space that contains equipment. Hierarchical decomposition of locations or neighbourhood relationships between locations are supported. Global positioning system data may be applied to specify the three-dimensional position.

The following application objects are used by the site UoF:

- Cartesian\_point;
- Cartesian\_coordinate\_space\_3d;
- General\_location\_relationship;
- Gis\_position;
- Hierarchical\_location\_relationship;
- Location;
- Location\_assignment;
- Location\_relationship;
- Neighbourhood\_location\_relationship.

#### 4.1.23 work\_management (W1)

The work\_management UoF specifies the concepts for activity specific, project specific, and contract specific information. This UoF also supports concepts to keep track of changes that result from various activities throughout the lifecycle of an electrotechnical system.

The following application objects are used by the work\_management UoF:

- Activity;

## **ISO 10303-212:2001(E)**

- Activity\_element;
- Activity\_method;
- Activity\_method\_assignment;
- Activity\_relationship;
- Contract;
- Organization\_in\_contract;
- Project;
- Project\_relationship;
- Work\_order;
- Work\_request.

### **4.2 Application objects**

This subclause specifies the application objects for the Electrotechnical design and installation application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

#### **4.2.1 Activity**

An Activity is the fact of achieving or accomplishing an action.

The data associated with an Activity are the following:

- activity\_type;
- actual\_end\_date;
- actual\_start\_date;
- chosen\_method;
- concerned\_organization;
- description;
- id;
- internal;
- planned\_end\_date;
- planned\_start\_date;
- requestor;
- resolved\_request;
- status;
- supplying\_organization.

#### **4.2.1.1 activity\_type**

The activity\_type specifies the kind of the Activity. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of activity\_type is one of the following:

- amendment;
- analysis;
- cancellation;
- delivery change;
- design change;
- design;
- mock-up creation;
- order;
- prototype building;
- rectification;
- restructuring;
- spare part creation;
- stop notice;
- testing;
- work definition.

NOTE See 4.2.1.1.1 - 4.2.1.1.14 for the definition of each predefined value for activity\_type.

### 4.2.1.1.1 amendment

amendment: An activity to add information to product data.

### 4.2.1.1.2 analysis

analysis: An activity to determine the behaviour of an item version under certain physical circumstances.

### 4.2.1.1.3 cancellation

cancellation: An activity to delete an item from the bill of material or to cancel the whole bill of material.

### 4.2.1.1.4 delivery change

delivery change: An activity to change an element delivery.

#### **4.2.1.1.5 design change**

design change: An activity to change the design or data such as geometry, master data, or properties of an item or an assembly.

#### **4.2.1.1.6 design**

design: An activity concerning the development of the design of an item or a product.

#### **4.2.1.1.7 mock-up creation**

mock-up creation: An activity to create an experimental model or replica of an item.

#### **4.2.1.1.8 order**

order: An activity to issue written direction to a manufacturer, tradesman, etc., to supply something.

#### **4.2.1.1.9 prototype creation**

prototype creation: An activity to manufacture the preliminary version of an item or product.

#### **4.2.1.1.10 rectification**

rectification: An activity to correct the data, documentation, or structure belonging to an item.

#### **4.2.1.1.11 restructuring**

restructuring: An activity to create a new structure or position within a bill of material without changing the data of the items.

#### **4.2.1.1.12 spare part creation**

spare part creation: An activity to design a spare part or to classify an item as a spare part.

#### **4.2.1.1.13 stop notice**

stop notice: An activity to stop the manufacturing process of an item.

#### **4.2.1.1.14 testing**

testing: An activity to test an item or a product.

#### **4.2.1.1.15 work definition**

work definition: An Activity to manage several sub-activities related to this Activity by an Activity\_ - relationship (see 4.2.5) with a 'relation\_type' of value 'decomposition'..

### **4.2.1.2 actual\_end\_date**

The actual\_end\_date specifies the date when the Activity was actually finished.

The actual\_end\_date need not be specified for a particular Activity.

## ISO 10303-212:2001(E)

See 4.3.3 for the application assertion.

### 4.2.1.3 actual\_start\_date

The actual\_start\_date specifies the date when the Activity was actually started.

The actual\_start\_date need not be specified for a particular Activity.

See 4.3.4 for the application assertion.

### 4.2.1.4 chosen\_method

The chosen\_method specifies the Activity method used to carry out the Activity.

The chosen\_method need not be specified for a particular Activity.

See 4.3.1 for the application assertion.

### 4.2.1.5 concerned\_organization

The concerned\_organization specifies the Organization (see 4.2.223) that is affected by the result of the Activity.

EXAMPLE The production site that has to change the process due to a design change activity is an example for concerned\_organization.

See 4.3.10 for the application assertion.

### 4.2.1.6 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Activity.

The description need not be specified for a particular Activity.

### 4.2.1.7 id

The id specifies the identifier of the Activity.

### 4.2.1.8 internal

The internal specifies whether or not the Activity is performed internally within the organization. An Activity that is not performed internally is accomplished externally by another organization.

The internal need not be specified for a particular Activity.

### 4.2.1.9 planned\_end\_date

The planned\_end\_date specifies the date when the Activity is or was supposed to be finished.

The planned\_end\_date need not be specified for a particular Activity.

Each `planned_end_date` may be one of the following: `Date_time` (see 4.2.79), `Duration` (see 4.2.126), or `Event_reference` (see 4.2.130).

See 4.3.5, 4.3.7, and 4.3.8 for the application assertions.

#### **4.2.1.10 `planned_start_date`**

The `planned_start_date` specifies the date when the Activity is or was supposed to be started.

The `planned_start_date` need not be specified for a particular Activity.

Each `planned_start_date` may be one of the following: `Date_time` (see 4.2.79) or `Event_reference` (see 4.2.130).

See 4.3.6 and 4.3.9 for the application assertions.

#### **4.2.1.11 `requestor`**

The `requestor` specifies the party that requested the Activity and the date the request was submitted.

The `requestor` need not be specified for a particular Activity.

See 4.3.2 for the application assertion.

#### **4.2.1.12 `resolved_request`**

The `resolved_request` specifies the `Work_request` (see 4.2.365) that is resolved by the Activity.

See 4.3.12 for the application assertion.

#### **4.2.1.13 `status`**

The `status` specifies the level of completion of the Activity.

EXAMPLE An example of `status` is 'open'.

The `status` need not be specified for a particular Activity.

#### **4.2.1.14 `supplying_organization`**

The `supplying_organization` specifies the Organization (see 4.2.223) that carries out the work.

See 4.3.11 for the application assertion.

### **4.2.2 `Activity_element`**

An `Activity_element` is an item of work that is part of an Activity (see 4.2.1).

## ISO 10303-212:2001(E)

The data associated with an Activity\_element are the following:

- associated\_activity;
- element;
- role.

### 4.2.2.1 associated\_activity

The associated\_activity specifies the Activity (see 4.2.1) the Activity\_element belongs to.

See 4.3.13 for the application assertion.

### 4.2.2.2 element

The element specifies the piece of product data that is under work.

Each element may be one of the following: Activity\_method (see 4.2.3), Activity\_relationship (see 4.2.5), Alternate\_item\_relationship (see 4.2.11), Alternate\_item\_relationship (see 4.2.11), Assembly\_component\_relationship (see 4.2.26), Cable\_pull\_information (see 4.2.33), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see 4.2.124), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Manufacturing\_configuration (see 4.2.198), Marking (see 4.2.199), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Requirement (see 4.2.285), Route (see 4.2.290), Route\_relationship (see 4.2.291), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327),



Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), or Terminal (see 4.2.338).

See 4.3.14, 4.3.15, 4.3.16, 4.3.17, 4.3.18, 4.3.19, 4.3.20, 4.3.21, 4.3.22, 4.3.23, 4.3.24, 4.3.25, 4.3.26, 4.3.27, 4.3.28, 4.3.29, 4.3.30, 4.3.31, 4.3.32, 4.3.33, 4.3.34, 4.3.35, 4.3.36, 4.3.37, 4.3.38, 4.3.39, 4.3.40, 4.3.41, 4.3.42, 4.3.43, 4.3.44, 4.3.45, 4.3.46, 4.3.47, 4.3.48, 4.3.49, 4.3.50, 4.3.51, 4.3.52, 4.3.53, 4.3.54, 4.3.55, 4.3.56, 4.3.57, 4.3.58, 4.3.59, 4.3.60, 4.3.61, 4.3.62, 4.3.63, 4.3.64, 4.3.65, 4.3.66, 4.3.67, 4.3.68, 4.3.69, 4.3.70, 4.3.71, 4.3.72, 4.3.73, 4.3.74, 4.3.75, 4.3.76, 4.3.77, 4.3.78, 4.3.79, 4.3.80, 4.3.81, 4.3.82, 4.3.83, 4.3.84, 4.3.85, 4.3.86, 4.3.87, 4.3.88, 4.3.89, 4.3.90, 4.3.91, 4.3.92, 4.3.93, 4.3.94, 4.3.95, 4.3.96, 4.3.97, 4.3.98, 4.3.99, 4.3.100, 4.3.101, 4.3.102, and 4.3.103 for the application assertions.

### 4.2.2.3 role

The role specifies the function that is performed by the Activity\_element in the context of the concerned Activity (see 4.2.1). The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- control;
- input;
- output.

NOTE See 4.2.2.3.1 - 4.2.2.3.3 for the definition of each permissible value for role.

#### 4.2.2.3.1 control

control: The referenced element is an object that has immediate influence on the Activity (see 4.2.1) performed.

EXAMPLE Design lots used in the design of a component that fulfils a given product function may be identified as Activity\_element objects that refer to this Function\_unit (see 4.2.148) in the role of 'control'.

#### 4.2.2.3.2 input

input: The referenced element serves as initial data for the associated Activity (see 4.2.1) object.

#### 4.2.2.3.3 output

output: The referenced element is the result of the associated Activity (see 4.2.1) object.

### 4.2.3 Activity\_method

An Activity\_method is a procedure that can be used to solve a problem.

## ISO 10303-212:2001(E)

The data associated with an Activity\_method are the following:

- consequence;
- description;
- name.

### 4.2.3.1 consequence

The consequence specifies the expected positive or negative effects of the application of a particular Activity\_method.

The consequence need not be specified for a particular Activity\_method.

### 4.2.3.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Activity\_method.

### 4.2.3.3 name

The name specifies the identifier of the Activity\_method.

## 4.2.4 Activity\_method\_assignment

An Activity\_method\_assignment is the relation that associates an Activity\_method (see 4.2.3) with a Work\_request (see 4.2.365). The associated Activity\_method (see 4.2.3) serves as a recommended method to resolve the tasks specified in the Work\_request (see 4.2.365).

The data associated with an Activity\_element (see 4.2.2) are the following:

- assigned\_method;
- assigned\_work\_request
- relation\_type.

### 4.2.4.1 assigned\_method

The assigned\_method specifies the Activity\_method (see 4.2.3).

See 4.3.104 for the application assertion.

### 4.2.4.2 assigned\_work\_request

The assigned\_work\_request specifies the Work\_request (see 4.2.365).

See 4.3.105 for the application assertion.

### 4.2.4.3 relation\_type

The `relation_type` specifies whether the specified `Activity_method` (see 4.2.3) may be used or not. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- non recommended method;
- recommended method.

NOTE See 4.2.4.3.1 - 4.2.4.3.2 for the definition of each predefined value for `relation_type`.

#### 4.2.4.3.1 non recommended method

non recommended method: The specified `Activity_method` (see 4.2.3) shall not be used in order to accomplish the specified `Work_request` (see 4.2.365).

#### 4.2.4.3.2 recommended method

recommended method: The specified `Activity_method` (see 4.2.3) may be used in order to accomplish the specified `Work_request` (see 4.2.365).

NOTE Several alternative recommended methods may exist.

### 4.2.5 Activity\_relationship

An `Activity_relationship` is the relation between two `Activity` (see 4.2.1) objects.

The data associated with an `Activity_relationship` are the following:

- description;
- related;
- relating;
- `relation_type`.

#### 4.2.5.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the `Activity_relationship`.

The description need not be specified for a particular `Activity_relationship`.

#### 4.2.5.2 related

The related specifies the second of the two `Activity` (see 4.2.1) objects related by an `Activity_relationship`.

NOTE The semantic of this attribute is defined by the attribute `relation_type`.

See 4.3.106 for the application assertion.

### 4.2.5.3 relating

The relating specifies the first of the two Activity (see 4.2.1) objects related by an Activity\_ - relationship.

NOTE The semantic of this attribute is defined by the attribute relation\_type.

See 4.3.107 for the application assertion.

### 4.2.5.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternative;
- decomposition;
- derivation;
- exclusiveness;
- precedence;
- sequence;
- simultaneity.

NOTE See 4.2.5.4.1 - 4.2.5.4.7 for the definition of each predefined value for relation\_type.

#### 4.2.5.4.1 alternative

alternative: The application object defines a relationship where the related Activity (see 4.2.1) may be used alternatively instead of the relating Activity (see 4.2.1).

#### 4.2.5.4.2 decomposition

decomposition: The Activity\_relationship defines a relationship where the related Activity (see 4.2.1) is one of the components into which the relating Activity (see 4.2.1) is broken down.

#### 4.2.5.4.3 derivation

derivation: The Activity\_relationship defines a relationship where the related Activity (see 4.2.1) is derived from the relating Activity (see 4.2.1).

EXAMPLE An Activity (see 4.2.1) controlled by a Work\_order (see 4.2.364) with work order type 'manufacturing release' can be derived from another Activity (see 4.2.1) controlled by another Work\_order (see 4.2.364) with work order type 'design release'.

#### 4.2.5.4.4 exclusiveness

exclusiveness: The application object defines a relationship where the relating and the related Activity (see 4.2.1) shall not have any overlap in time of execution.

#### 4.2.5.4.5 precedence

precedence: The Activity\_relationship defines a relationship where the related Activity (see 4.2.1) has higher priority than the relating Activity (see 4.2.1).

#### 4.2.5.4.6 sequence

sequence: The Activity\_relationship defines a relationship where the relating Activity (see 4.2.1) shall be completed before the related Activity (see 4.2.1) starts.

#### 4.2.5.4.7 simultaneity

simultaneity: The Activity\_relationship defines a relationship where the related Activity (see 4.2.1) is carried out simultaneously with the relating Activity (see 4.2.1).

### 4.2.6 Address

An Address is the place where people and organizations are located.

## ISO 10303-212:2001(E)

The data associated with an Address are the following:

- country;
- email\_address;
- fax\_number;
- internal\_location;
- postal\_box;
- postal\_code;
- region;
- street;
- street\_number;
- telephone\_number;
- telex\_number;
- town.

### 4.2.6.1 country

The country specifies the name of a nation.

The country need not be specified for a particular Address.

### 4.2.6.2 email\_address

The email\_address specifies the sequence of characters that make up the appropriate address for electronic mail.

The email\_address need not be specified for a particular Address.

### 4.2.6.3 fax\_number

The fax\_number specifies the number at which facsimiles can be received.

The fax\_number need not be specified for a particular Address.

### 4.2.6.4 internal\_location

The internal\_location specifies the organization-defined address for internal mail delivery.

The internal\_location need not be specified for a particular Address.

### 4.2.6.5 postal\_box

The postal\_box specifies the number of the appropriate post office box.

The postal\_box need not be specified for a particular Address.

#### **4.2.6.6 postal\_code**

The postal\_code specifies the code that is used by the local postal service.

The postal\_code need not be specified for a particular Address.

#### **4.2.6.7 region**

The region specifies the name of the area.

The region need not be specified for a particular Address.

#### **4.2.6.8 street**

The street specifies the name of the road.

The street need not be specified for a particular Address.

#### **4.2.6.9 street\_number**

The street\_number specifies the number of the building in a street.

The street\_number need not be specified for a particular Address.

#### **4.2.6.10 telephone\_number**

The telephone\_number specifies the number at which telephone calls can be received.

The telephone\_number need not be specified for a particular Address.

#### **4.2.6.11 telex\_number**

The telex\_number specifies the number at which telex messages can be received.

The telex\_number need not be specified for a particular Address.

#### **4.2.6.12 town**

The town specifies the name of a city.

The town need not be specified for a particular Address.

### **4.2.7 Aggregated\_value**

An Aggregated\_value is a type of Data\_element\_value (see 4.2.76) that is used to organize the values of a Data\_element (see 4.2.70) as a list or as an array. The meaning of the order of the associated values shall be defined in the associated Data\_element\_definition (see 4.2.72).

## ISO 10303-212:2001(E)

NOTE Multidimensional arrays are created through nesting Aggregated\_value objects.

The data associated with an Aggregated\_value are the following:

— member\_definition.

### 4.2.7.1 member\_definition

The member\_definition specifies the constituents of Aggregated\_value.

See 4.3.108 for the application assertion.

### 4.2.8 Alias\_designation

An Alias\_designation is the mechanism to associate an object with an additional designation that is used to identify the object of interest in a different context, either in another Organization (see 4.2.223), or in some other context. The Alias\_designation allows to specify structured labels where the semantic of each composite can be identified. The scope of the Alias\_designation shall be specified either by the attribute 'alias\_scope' or by the attribute 'description'.

NOTE The Alias\_designation allows to assign an alias name that is not just a simple string but a designation equipped with the information provided by the Object\_designation (see 4.2.217) object.

EXAMPLE An relais may be designated as '=A00-K1' in the context of the supplier and as '=N005.A00-K1' in the context of the customer.

The data associated with an Alias\_designation are the following:

— alias\_extended\_designation;

— alias\_scope;

— description;

— is\_applied\_to.

#### 4.2.8.1 alias\_extended\_designation

The alias\_extended\_designation specifies the designator. The designator may be a structured label.

See 4.3.115 for the application assertion.

#### 4.2.8.2 alias\_scope

The alias\_scope specifies the Organization (see 4.2.223) in which the Alias\_designation is valid.

The alias\_scope need not be specified for a particular Alias\_designation.

See 4.3.116 for the application assertion.

#### 4.2.8.3 description



The description specifies the kind of the Alias\_designation.

EXAMPLE The description may be 'preliminary designation'.

The description need not be specified for a particular Alias\_designation.

#### 4.2.8.4 is\_applied\_to

The is\_applied\_to specifies the object that has an Alias\_designation.

Each is\_applied\_to may be one of the following: Device (see 4.2.88), Document\_representation (see 4.2.110), Drawing (see 4.2.119), Drawing\_sheet (see 4.2.122), Function\_unit (see 4.2.148), Location (see 4.2.192), Port (see 4.2.247), Product\_component (see 4.2.265), Signal (see 4.2.309), Technical\_system (see 4.2.336), or Terminal (see 4.2.338).

See 4.3.109, 4.3.110, 4.3.111, 4.3.112, 4.3.113, 4.3.114, 4.3.117, 4.3.118, 4.3.119, 4.3.120, and 4.3.121 for the application assertions.

#### 4.2.9 Alias\_identification

An Alias\_identification is the mechanism to associate an object with an additional identifier that is used to identify the object of interest in a different context, either in another Organization (see 4.2.223), or in some other context. The scope of the Alias\_identification shall be specified either by the attribute 'alias\_scope' or by the attribute 'description'.

NOTE The identifier may be used to identify the object of interest in a different context, either in another Organization (see 4.2.223), or in some other context.

EXAMPLE A book may have a unique document id (ISBN) and an Alias\_identification as an inventory number in the context of the inventory of a company.

The data associated with an Alias\_identification are the following:

- alias\_id;
- alias\_scope;
- description;
- is\_applied\_to.

##### 4.2.9.1 alias\_id

The alias\_id specifies the identifier used in the context specified by alias\_scope, description, or both.

##### 4.2.9.2 alias\_scope

The alias\_scope specifies the Organization (see 4.2.223) in which the Alias\_identification is valid.

The alias\_scope need not be specified for a particular Alias\_identification.

See 4.3.149 for the application assertion.

### 4.2.9.3 description

The description specifies the kind of the Alias\_identification.

EXAMPLE The description may be 'inventory number'.

The description need not be specified for a particular Alias\_identification.

### 4.2.9.4 is\_applied\_to

The is\_applied\_to specifies the object that has an Alias\_identification.

Each is\_applied\_to may be one of the following: Approval\_status (see 4.2.25), Classification\_attribute (see 4.2.47), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Component\_colour (see 4.2.53), Connectivity\_definition (see 4.2.61), Data\_element\_definition (see 4.2.72), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Document (see 4.2.101), Document\_representation (see 4.2.110), Document\_type\_property (see 4.2.113), Document\_version (see 4.2.114), Drawing (see 4.2.119), Drawing\_sheet (see 4.2.122), Function\_definition (see 4.2.145), Function\_unit (see 4.2.148), Functional\_connectivity\_definition (see 4.2.152), Functionality (see 4.2.155), General\_classification (see 4.2.156), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_version (see 4.2.182), Location (see 4.2.192), Node (see 4.2.208), Notification (see 4.2.213), Organization (see 4.2.223), Path (see 4.2.232), Path\_node (see 4.2.233), Physical\_instance (see 4.2.243), Port (see 4.2.247), Process\_variable (see 4.2.260), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Requirement (see 4.2.285), Route (see 4.2.290), Section (see 4.2.296), Section\_interface (see 4.2.298), Security\_level (see 4.2.302), Signal (see 4.2.309), Specification (see 4.2.323), Specification\_category (see 4.2.324), Technical\_system (see 4.2.336), or Terminal (see 4.2.338).

See 4.3.122, 4.3.123, 4.3.124, 4.3.125, 4.3.126, 4.3.127, 4.3.128, 4.3.129, 4.3.130, 4.3.131, 4.3.132, 4.3.133, 4.3.134, 4.3.135, 4.3.136, 4.3.137, 4.3.138, 4.3.139, 4.3.140, 4.3.141, 4.3.142, 4.3.143, 4.3.144, 4.3.145, 4.3.146, 4.3.147, 4.3.148, 4.3.150, 4.3.151, 4.3.152, 4.3.153, 4.3.154, 4.3.155, 4.3.156, 4.3.157, 4.3.158, 4.3.159, 4.3.160, 4.3.161, 4.3.162, 4.3.163, 4.3.164, 4.3.165, 4.3.166, and 4.3.167 for the application assertions.

### 4.2.10 Alias\_version

An Alias\_version is a particular version of the object to which the associated alias identifier applies.

NOTE 1 The scope of the Alias\_version is specified either through the 'alias\_scope' attribute of the associated Alias\_identification (see 4.2.9) object or Alias\_designation (see 4.2.8) object.

NOTE 2 An Alias\_version may be applied only if the object of interest may carry an optional or mandatory version. If so, an Alias\_version may be assigned even if the object of interest does not yet have a version assigned.

The data associated with an Alias\_version are the following:

- associated\_alias\_id;
- version\_id.

#### 4.2.10.1 associated\_alias\_id

The associated\_alias\_id specifies the identifier.

Each associated\_alias\_id may be one of the following: Alias\_designation (see 4.2.8) or Alias\_identification (see 4.2.9).

See 4.3.168 and 4.3.169 for the application assertions.

#### 4.2.10.2 version\_id

The version\_id specifies the version of the object as known in the context of the alias identifier.

#### 4.2.11 Alternate\_item\_relationship

An Alternate\_item\_relationship is the relationship between two Item (see 4.2.178) objects specifying that all versions of the two related Item (see 4.2.178) objects are interchangeable independent from their context of use.

NOTE 1 Interchangeability usually refers to form, fit, function, and quality. Additional properties, such as performance, noise, endurance, or reliability, may also be considered as a prerequisite for the interchangeability.

NOTE 2 If more than one Alternate\_item\_relationship is in place, the union of the attributes fulfilled\_requirements is the criteria for the interchangeability.

EXAMPLE The Alternate\_item\_relationship indicates that an amplifier XX741 may be replaced by an amplifier YY741 from another manufacturer.

The two Item (see 4.2.178) objects are interchangeable in both directions.

## ISO 10303-212:2001(E)

The data associated with an Alternate\_item\_relationship are the following:

- alternate;
- base;
- fulfilled\_requirements.

### 4.2.11.1 alternate

The alternate specifies the Item (see 4.2.178) that may be used in place of the base Item (see 4.2.178).

See 4.3.170 for the application assertion.

### 4.2.11.2 base

The base specifies the Item (see 4.2.178) for which another Item (see 4.2.178) may be used as an alternate.

See 4.3.171 for the application assertion.

### 4.2.11.3 fulfilled\_requirements

The fulfilled\_requirements specifies the word or group of words describing the requirements that are covered by both the base and the alternate and are, therefore, the basis for the statement of interchangeability.

## 4.2.12 Alternative\_solution

An Alternative\_solution is a type of Complex\_product (see 4.2.51) that is the design of one of potentially many mutually exclusive implementation options. An Alternative\_solution may refer directly to the object to be implemented or through another Alternative\_solution, in which case, it serves as a refinement of that Alternative\_solution.

The data associated with an Alternative\_solution are the following:

- base\_element

### 4.2.12.1 base\_element

The base\_element specifies the physical or functional object, for which the Alternative\_solution provides a design alternative. All Alternative\_solution objects for the same base element are mutually exclusive.

Each base\_element may be one of the following: Alternative\_solution, Function\_definition (see 4.2.145), Product\_component (see 4.2.265), or Single\_function\_unit (see 4.2.315).

See 4.3.172, 4.3.173, 4.3.174, and 4.3.175 for the application assertions.

### 4.2.13 Angular\_dimension

An `Angular_dimension` is a type of `Dimension` (see 4.2.93) that is the graphical presentation of a value of the angle between two elements that converge on a common point or line.

The data associated with an `Angular_dimension` are the following:

- component;
- extent.

#### 4.2.13.1 component

The component specifies the projection lines that show the extension of the points, lines, or surfaces that bound the measurement.

See 4.3.177 for the application assertion.

#### 4.2.13.2 extent

The extent specifies the dimension line that graphically presents where the dimension value applies.

See 4.3.176 for the application assertion.

### 4.2.14 Annotation\_curve

An `Annotation_curve` is a type of `Annotation_element` (see 4.2.15) that is a two-dimensional trimmed curve used only to annotate a drawing or a draughting shape model and that is defined in the coordinate system in which it is used.

The data associated with an `Annotation_curve` are the following:

- assigned\_appearance.

#### 4.2.14.1 assigned\_appearance

The `assigned_appearance` specifies the presentation aspects of an `Annotation_curve`.

See 4.3.178 for the application assertion.

### 4.2.15 Annotation\_element

An `Annotation_element` is a type of `Draughting_annotation` (see 4.2.116) that is the lowest level discrete element that can either serve as an annotation itself, or be used as a constituent of other annotations.

Each `Annotation_element` is either an `Annotation_curve` (see 4.2.14), an `Annotation_subfigure` (see 4.2.17), an `Annotation_symbol` (see 4.2.20), a `Fill_area` (see 4.2.139), a `Schematic_node` (see 4.2.294), or a `Text` (see 4.2.342).

### 4.2.16 Annotation\_placed\_annotation

## ISO 10303-212:2001(E)

An `Annotation_placed_annotation` is a type of `Draughting_annotation` (see 4.2.116) that is located in the coordinate system of a symbol or subfigure.

### 4.2.17 `Annotation_subfigure`

An `Annotation_subfigure` is a type of `Annotation_element` (see 4.2.15) that is the visual instance of an `Annotation_subfigure_definition` (see 4.2.18) located within the coordinate system of a drawing sheet, a drawing view, a draughting model, or another subfigure.

The data associated with an `Annotation_subfigure` are the following:

- definition;
- position;
- rotation;
- scale.

#### 4.2.17.1 `definition`

The definition specifies the template from which the `Annotation_subfigure` is derived.

See 4.3.179 for the application assertion.

#### 4.2.17.2 `position`

The position specifies the location of the origin of the coordinate system in which the subfigure is defined relative to the origin of the coordinate system into which the subfigure is being placed.

See 4.3.180 for the application assertion.

#### 4.2.17.3 `rotation`

The rotation specifies the angle, measured counter-clockwise, between the horizontal axis of the coordinate system in which the subfigure is defined and the horizontal axis of the coordinate system into which the subfigure is being placed.

#### 4.2.17.4 `scale`

The scale specifies the ratio between the size of the subfigure as defined and the size of the subfigure as presented.

### 4.2.18 `Annotation_subfigure_definition`

An `Annotation_subfigure_definition` is a collection of defined annotation elements, along with their placements, in a coordinate space.

The data associated with an `Annotation_subfigure_definition` are the following:

- `blanking_box`;
- `coordinate_space`;
- `name`.

#### 4.2.18.1 `blanking_box`

The `blanking_box` specifies an area that the `Annotation_subfigure_definition` occupies and is used to suppress the visual presentation of all other elements that are within this area.

The `blanking_box` need not be specified for a particular `Annotation_subfigure_definition`.

See 4.3.182 for the application assertion.

#### 4.2.18.2 `coordinate_space`

The `coordinate_space` specifies the coordinate system that describes the two-dimensional space in which the constituents of the `Annotation_subfigure_definition` are located.

See 4.3.181 for the application assertion.

#### 4.2.18.3 `name`

The `name` specifies the identifier of the `Annotation_subfigure_definition`.

The `name` need not be specified for a particular `Annotation_subfigure_definition`.

### 4.2.19 `Annotation_subfigure_definition_element`

An `Annotation_subfigure_definition_element` is an annotation that is used as a constituent of a subfigure definition.

The data associated with an `Annotation_subfigure_definition_element` are the following:

- `annotation_layers`;
- `annotation_visibility`;
- `containing_definition`;
- `used_annotation`.

#### 4.2.19.1 `annotation_layers`

The `annotation_layers` specifies the draughting layers that contain the annotation.

See 4.3.185 for the application assertion.

#### 4.2.19.2 `annotation_visibility`

## ISO 10303-212:2001(E)

The `annotation_visibility` specifies whether or not each element of a subfigure is visible.

See 4.3.186 for the application assertion.

### 4.2.19.3 `containing_definition`

The `containing_definition` specifies the subfigure definition in which the annotation is placed.

See 4.3.184 for the application assertion.

### 4.2.19.4 `used_annotation`

The `used_annotation` specifies the annotation that is an element of the subfigure.

See 4.3.183 for the application assertion.

### 4.2.20 `Annotation_symbol`

An `Annotation_symbol` is a type of `Annotation_element` (see 4.2.15) that is the presentation of a symbol definition that is either externally defined, predefined, or defined explicitly as a combination of annotation elements. The `Annotation_symbol` is located within the coordinate system of a drawing sheet, drawing view, or another symbol.

Each `Annotation_symbol` is either an `Externally_defined_symbol` (see 4.2.135), a `Predefined_symbol` (see 4.2.255), or a `User_defined_symbol` (see 4.2.354).

The data associated with an `Annotation_symbol` are the following:

- `blanking_box`;
- `overriding_colour`;
- `position`;
- `rotation`;
- `scale`.

#### 4.2.20.1 `blanking_box`

The `blanking_box` specifies an area that the `Annotation_element` (see 4.2.15) occupies and is used to suppress the visual presentation of all other elements that are within this area.

The `blanking_box` need not be specified for a particular `Annotation_symbol`.

See 4.3.189 for the application assertion.

#### 4.2.20.2 `overriding_colour`

The `overriding_colour` specifies the colour definition that overrides the appearance characteristics already assigned to the elements of the symbol.



The overriding\_colour need not be specified for a particular Annotation\_symbol.

See 4.3.187 for the application assertion.

### 4.2.20.3 position

The position specifies the location of the origin of the coordinate system in which the symbol is defined relative to the origin of the coordinate system into which the symbol is being placed.

See 4.3.188 for the application assertion.

### 4.2.20.4 rotation

The rotation specifies the angle, measured counter-clockwise, between the positive x-axis of the coordinate system in which the symbol is defined and the positive x-axis of the coordinate system into which the symbol is being placed.

### 4.2.20.5 scale

The scale specifies the ratio between the size of the symbol as defined and the size of the symbol as presented. The scale in the x-coordinate need not equal the scale in the y-coordinate.

## 4.2.21 Appearance

An Appearance is a collection of visual characteristics that govern the presentation of geometric elements or annotation elements.

Each Appearance is either a Curve\_appearance (see 4.2.68), a Fill\_area\_appearance (see 4.2.140), or a Text\_appearance (see 4.2.343).

## 4.2.22 Application\_context

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

The data associated with an Application\_context are the following:

- application\_domain;
- description;
- life\_cycle\_stage.

### 4.2.22.1 application\_domain

The application\_domain is the identification of the discipline for which the product data is relevant.

EXAMPLE Examples for application\_domain are 'electrical design' or 'engineering analysis'.

### 4.2.22.2 description

## ISO 10303-212:2001(E)

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Application\_context.

The description need not be specified for a particular Application\_context.

### 4.2.22.3 life\_cycle\_stage

The life\_cycle\_stage is the specification of the general stage in the product life cycle to which the product data belong. The value is either user defined or predefined.

The predefined value of life\_cycle\_stage is one of the following:

- conceptual design;
- development;
- implementation;
- operation;
- test.

NOTE See 4.2.22.3.1 - 4.2.22.3.5 for the definition of each predefined value for life\_cycle\_stage.

#### 4.2.22.3.1 conceptual design

conceptual design: The object that refers the Application\_context object is in the stage of transforming the customer's requirements into a draft design.

#### 4.2.22.3.2 development

development: The object that refers the Application\_context object is in the stage of transforming the conceptual design into an implementable system.

#### 4.2.22.3.3 implementation

implementation: The object that refers the Application\_context object is in the stage of realization.

#### 4.2.22.3.4 operation

operation: The object that refers the Application\_context object is in the stage of productive use.

#### 4.2.22.3.5 test

test: The object that refers the Application\_context object is in the stage of verification whether it satisfies the specified requirements.

### 4.2.23 Approval

An Approval is a judgement concerning the quality of product data that are subject of the Approval. An Approval represents a statement made by technical personnel or management personnel if certain requirements are met.

The data associated with an Approval are the following:

- actual\_date;
- is\_applied\_to;
- is\_approved\_by;
- level;
- planned\_date;
- scope;
- status.

#### 4.2.23.1 actual\_date

The actual\_date specifies the date when the Approval was actually performed. If this information is absent, the approval has not yet occurred, i.e., it is pending.

The actual\_date need not be specified for a particular Approval.

See 4.3.220 for the application assertion.

#### 4.2.23.2 is\_applied\_to

The is\_applied\_to specifies the object to which the Approval is assigned.

Each is\_applied\_to may be one of the following: Activity (see 4.2.1), Activity\_element (see 4.2.2), Activity\_method\_assignment (see 4.2.4), Activity\_relationship (see 4.2.5), Alternate\_item\_relationship (see 4.2.11), Alternate\_item\_relationship (see 4.2.11), Assembly\_component\_relationship (see 4.2.26), Cable\_pull\_information (see 4.2.33), Certification (see 4.2.38), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_association (see 4.2.46), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_allocation (see 4.2.60), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Contract (see 4.2.63), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see 4.2.124), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship

## ISO 10303-212:2001(E)

(see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Manufacturing\_configuration (see 4.2.198), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Route (see 4.2.290), Route\_relationship (see 4.2.291), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.190, 4.3.191, 4.3.192, 4.3.193, 4.3.194, 4.3.195, 4.3.197, 4.3.198, 4.3.199, 4.3.200, 4.3.201, 4.3.202, 4.3.203, 4.3.204, 4.3.205, 4.3.206, 4.3.207, 4.3.208, 4.3.209, 4.3.210, 4.3.211, 4.3.212, 4.3.213, 4.3.214, 4.3.215, 4.3.216, 4.3.217, 4.3.218, 4.3.222, 4.3.223, 4.3.224, 4.3.225, 4.3.226, 4.3.227, 4.3.228, 4.3.229, 4.3.230, 4.3.231, 4.3.232, 4.3.233, 4.3.234, 4.3.235, 4.3.236, 4.3.237, 4.3.238, 4.3.239, 4.3.240, 4.3.241, 4.3.242, 4.3.243, 4.3.244, 4.3.245, 4.3.246, 4.3.247, 4.3.248, 4.3.249, 4.3.250, 4.3.251, 4.3.252, 4.3.253, 4.3.254, 4.3.255, 4.3.256, 4.3.257, 4.3.258, 4.3.259, 4.3.260, 4.3.261, 4.3.262, 4.3.264, 4.3.265, 4.3.266, 4.3.267, 4.3.268, 4.3.269, 4.3.270, 4.3.271, 4.3.272, 4.3.273, 4.3.274, 4.3.275, 4.3.276, 4.3.277, 4.3.278, 4.3.279, 4.3.280, 4.3.281, 4.3.282, 4.3.283, 4.3.284, 4.3.285, 4.3.286, 4.3.287, 4.3.288, 4.3.289, 4.3.290, 4.3.291, 4.3.292, 4.3.293, 4.3.294, 4.3.295, 4.3.296, 4.3.297, 4.3.298, 4.3.299, and 4.3.300 for the application assertions.

### 4.2.23.3 is\_approved\_by

The is\_approved\_by specifies the Person (see 4.2.237) or Organization (see 4.2.223) responsible for the Approval and the date when the authorization was granted.

See 4.3.219 for the application assertion.

### 4.2.23.4 level

The level indicates the kind of activity that may be performed. The value is either user defined or predefined.

The predefined value of level is one of the following:

- disposition;
- equipment order;
- planning.

NOTE See 4.2.23.4.1 - 4.2.23.4.3 for the definition of each predefined value for level.

#### **4.2.23.4.1 disposition**

disposition: The approved item is approved for series production.

#### **4.2.23.4.2 equipment order**

equipment order: The approved item has reached the status in which changes are subject to a defined change process so that tools and other equipment for the production may be ordered.

#### **4.2.23.4.3 planning**

planning: The approved item is technically complete and has reached the status sufficiently stable so that other designs may be based on it.

The level need not be specified for a particular Approval.

#### **4.2.23.5 planned\_date**

The planned\_date specifies the date when the Approval is or was supposed to be performed.

The planned\_date need not be specified for a particular Approval.

See 4.3.221 for the application assertion.

#### **4.2.23.6 scope**

The scope specifies the set of Organization (see 4.2.223) objects for which the Approval is valid.

See 4.3.263 for the application assertion.

#### **4.2.23.7 status**

The status specifies the judgement made about the product data that is the subject of this Approval.

See 4.3.196 for the application assertion.

#### **4.2.24 Approval\_relationship**

An Approval\_relationship is the relation between two Approval (see 4.2.23) objects.

## ISO 10303-212:2001(E)

The data associated with an Approval\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.24.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Approval\_relationship.

The description need not be specified for a particular Approval\_relationship.

### 4.2.24.2 related

The related specifies the second of the two Approval (see 4.2.23) objects related by the Approval\_relationship.

NOTE The semantic of this attribute is defined by the attribute relation\_type.

See 4.3.301 for the application assertion.

### 4.2.24.3 relating

The relating specifies the first of the two Approval (see 4.2.23) objects related by the Approval\_relationship.

NOTE The semantic of this attribute is defined by the attribute relation\_type.

See 4.3.302 for the application assertion.

### 4.2.24.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The value of `relation_type` is one of the following:

- decomposition;
- sequence;
- precedence.

NOTE See 4.2.24.4.1 - 4.2.24.4.3 for the definition of each predefined value for `relation_type`.

#### 4.2.24.4.1 decomposition

decomposition: The `Approval_relationship` defines a relationship where the related `Approval` (see 4.2.23) is one of the components into which the relating `Approval` (see 4.2.23) is broken down.

#### 4.2.24.4.2 sequence

sequence: The `Approval_relationship` defines a relationship where the relating `Approval` (see 4.2.23) shall be completed before the related `Approval` (see 4.2.23) is given.

#### 4.2.24.4.3 precedence

precedence: The `Approval_relationship` defines a relationship where the related `Approval` (see 4.2.23) has higher priority than the relating `Approval` (see 4.2.23).

### 4.2.25 Approval\_status

An `Approval_status` is the state of acceptance of some product data.

The value is either user defined or predefined. The data associated with an `Approval_status` are the following:

- `status_name`;
- `used_classification_system`.

#### 4.2.25.1 status\_name

The `status_name` specifies the word or abbreviation that is used to refer to the `Approval_status`. The value is either user defined or predefined.

The predefined value of `status_name` is one of the following:

- approved;
- disapproved;
- withdrawn.

NOTE See 4.2.25.1.1 - 4.2.25.1.3 for the definition of each permissible value for `status_name`.

##### 4.2.25.1.1 approved

## **ISO 10303-212:2001(E)**

approved: The fact that the judgement on the referenced element is positive.

### **4.2.25.1.2 disapproved**

disapproved: The fact that the judgement on the referenced element is negative.

### **4.2.25.1.3 withdrawn**

withdrawn: The fact that the referenced element was cancelled.

## **4.2.25.2 used\_classification\_system**

The used\_classification\_system specifies the Classification\_system (see 4.2.48) that contains the information about how to interpret the Approval (see 4.2.23) status.

The used\_classification\_system need not be specified for a particular Approval status.

See 4.3.303 for the application assertion.

## **4.2.26 Assembly\_component\_relationship**

An Assembly\_component\_relationship is the relation between an Assembly\_definition (see 4.2.27) and a Device (see 4.2.88). The Device (see 4.2.88) represents a constituent of the assembly or package.



NOTE 1 The constituent may also be an assembly.

NOTE 2 The same instance of Device (see 4.2.88) may be a constituent of more than one assembly defined through the Design\_discipline\_item\_definition (see 4.2.86) object.

NOTE 3 An assembly may be regarded to be a package of things that belong together for some purpose. Following that idea, devices may belong to more than one of such a package. This part of ISO 10303 does not constrain the Assembly\_component\_relationship between the packages. One may be a subset of another, they may be disjoint, or they may partially overlap.

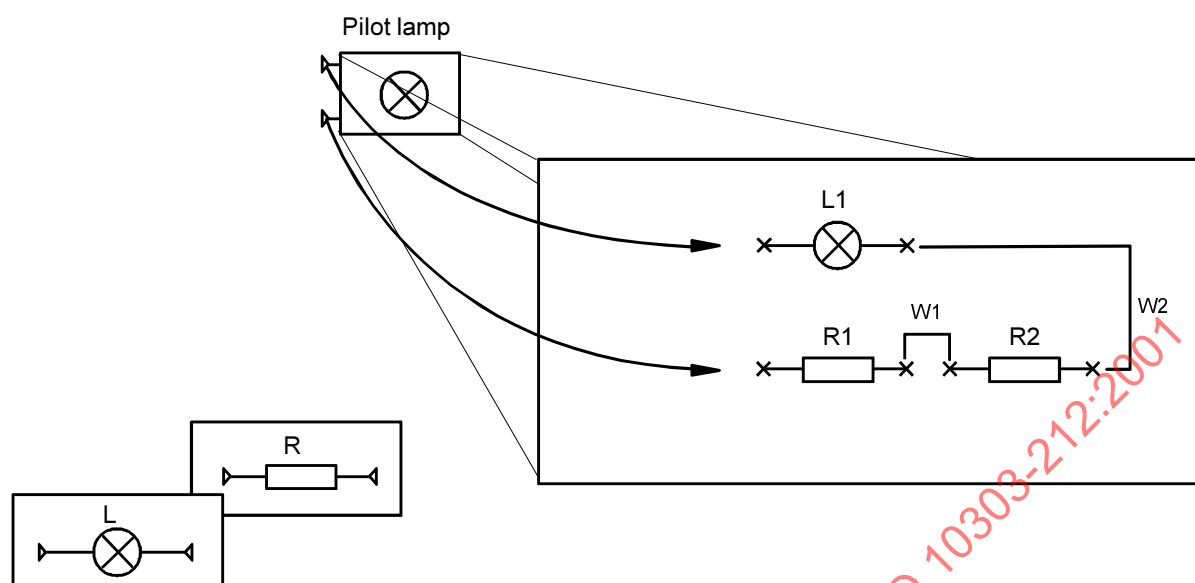
EXAMPLE 1 The power supply cord of an overhead projector may belong at the same time to the sales package of the projector and to the package of power supply cords that are currently in stock.

EXAMPLE 2 In Figure 7, the structure of a simple assembly is given to show how this part of ISO 10303 is to be used to specify hierarchically composed pieces of equipment. A pilot lamp is shown which is composed of the lamp L1 and the resistors R1 and R2. The connections W1 and W2 perform the internal connectivity. Templates for the lamp and the resistors exist (L and R) that characterize the properties of the components, which are independent from the context of usage, e.g., specification sheet parameters.

Two instances of Design\_discipline\_item\_definition (see 4.2.86) represent the templates R and L. The template L has two instances of Interface\_terminal (see 4.2.174) assigned to it, in order to characterize its two connect nodes. The connect nodes are indicated in the figure by triangles. The same applies to template R.

L1 is an occurrence of L, and R1 and R2 are occurrences of R. L1, R2, and R3 are represented by Device (see 4.2.88) objects which are associated by their 'definition' attributes with the Design\_discipline\_item\_definition (see 4.2.86) representing L respectively R. The connect nodes of L1, R1, and R2 are characterized through Terminal (see 4.2.338) objects. Figure 7 shows the six instances of the object Terminal (see 4.2.338) (indicated by an 'x'). Each of the Terminal (see 4.2.338) object is associated with an Interface\_terminal (see 4.2.174) through the 'associated\_interface\_terminal' attribute. The connectivity is performed by the Connection (see 4.2.59) objects W1 and W2 that link the appropriate Terminal (see 4.2.338) objects.

An instance of Assembly\_definition (see 4.2.27), which is a subtype of Design\_discipline\_item\_definition (see 4.2.86), specifies the pilot lamp. Three instances of Next\_higher\_assembly (see 4.2.207) express that this assembly is made from L1, R1, and R2. To express the association between the Interface\_terminal (see 4.2.174) objects of the pilot lamp and its appropriate counterpart at the next lower level of detail, the 'uses' attribute of Interface\_terminal (see 4.2.174) is employed. The curved arrows in the figure symbolize these associations.



**Figure 7 - Hierarchical composition**

Each `Assembly_component_relationship` is either a `Next_higher_assembly` (see 4.2.207) or a `Promissory_usage` (see 4.2.274).

The data associated with an `Assembly_component_relationship` are the following:

- placement;
- related;
- relating.

#### 4.2.26.1 placement

The placement specifies the geometrical transformation that is used to calculate the spatial position of the constituent within the assembly.

The placement need not be specified for a particular `Assembly_component_relationship`.

#### 4.2.26.2 related

The related specifies the Device (see 4.2.88) that acts as a component.

See 4.3.305 for the application assertion.

#### 4.2.26.3 relating

The relating specifies the `Assembly_definition` (see 4.2.27) that has subordinate constituents.

See 4.3.304 for the application assertion.

#### 4.2.27 Assembly\_definition

An `Assembly_definition` is a type of `Design_discipline_item_definition` (see 4.2.86) that is the definition of an `Item_version` (see 4.2.182) that contains other subordinate `Item_version` (see 4.2.182) objects.

NOTE An `Assembly_definition` can be used to define items such as an assembled module or a procurement package.

EXAMPLE A procurement package consists of a power transformer together with its mounting material.

The data associated with an `Assembly_definition` are the following:

— `assembly_type`.

#### 4.2.27.1 `assembly_type`

The `assembly_type` specifies the kind of the `Assembly_definition`.

EXAMPLE 'functional assembly', 'manufacturing assembly', and 'design assembly' are examples of an `assembly_type`.

The `assembly_type` need not be specified for a particular `Assembly_definition`.

### 4.2.28 `Assembly_substitute_relationship`

An `Assembly_substitute_relationship` is a relationship that indicates that one `Assembly_component_relationship` (see 4.2.26) may be substituted for another `Assembly_component_relationship` (see 4.2.26). The subjects of the substitution are the related `Device` (see 4.2.88) objects of both `Assembly_component_relationship` (see 4.2.26) objects. The relating `Assembly_definition` (see 4.2.27) shall be the same in both `Assembly_component_relationship` (see 4.2.26) objects.

The data associated with an `Assembly_substitute_relationship` are the following:

— `base`;  
 — `description`;  
 — `substitute`.

#### 4.2.28.1 `base`

The `base` specifies the `Assembly_component_relationship` (see 4.2.26) that is replaceable.

See 4.3.306 for the application assertion.

#### 4.2.28.2 `description`

The `description` specifies additional information about the `Assembly_substitute_relationship`.

The `description` need not be specified for a particular `Assembly_substitute_relationship`.

#### 4.2.28.3 `substitute`

## ISO 10303-212:2001(E)

The substitute specifies the `Assembly_component_relationship` (see 4.2.26) that may be used instead of the base `Assembly_component_relationship` (see 4.2.26).

See 4.3.307 for the application assertion.

### 4.2.29 `Binary_value`

A `Binary_value` is a sequence of digits that may have the value 0 or 1.

NOTE `Binary_value` allows the assignment of values of data type 'binary' to data elements.

EXAMPLE An example for the use of binary values is the description of the configuration of measured-value logger boards.

The data associated with a `Binary_value` are the following:

— `value_of_binary_value`.

#### 4.2.29.1 `value_of_binary_value`

The `value_of_binary_value` specifies a sequence of binary digits.

### 4.2.30 `Body_breadth`

A `Body_breadth` is the measured distance from side to side of a component in the y-axis.

The data associated with a `Body_breadth` are the following:

— `value_of_body_breadth`.

#### 4.2.30.1 `value_of_body_breadth`

The `value_of_body_breadth` specifies the side to side distance value in the y-axis.

See 4.3.308 for the application assertion.

### 4.2.31 `Body_height`

A `Body_height` is the measured distance from side to side of a component in the z-axis.

The data associated with an `Body_height` are the following:

— `value_of_body_height`.

#### 4.2.31.1 `value_of_body_height`

The `value_of_body_height` specifies the side to side distance value in the z-axis.

See 4.3.309 for the application assertion.

### 4.2.32 `Body_length`

A `Body_length` is the measured distance from side to side of a component in the x-axis.

The data associated with a `Body_length` are the following:

— `value_of_body_length`.

#### **4.2.32.1 value\_of\_body\_length**

The `value_of_body_length` specifies the side to side distance value in the x-axis.

See 4.3.310 for the application assertion.

#### **4.2.33 Cable\_pull\_information**

A `Cable_pull_information` is a collection of facts about the installation of a cable or wire harness on a segment of its path.

NOTE Additional information can be provided by assigning `Data_element` (see 4.2.70) objects to `Cable_pull_information`.

The data associated with a `Cable_pull_information` are the following:

- `associated_object`;
- `description`;
- `id`;
- `version_id`.

##### **4.2.33.1 associated\_object**

The `associated_object` specifies the equipment items to which the pulling information applies.

See 4.3.311 for the application assertion.

##### **4.2.33.2 description**

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Cable_pull_information`.

The `description` need not be specified for a particular `Cable_pull_information`.

##### **4.2.33.3 id**

The `id` specifies the identifier of the `Cable_pull_information`.

##### **4.2.33.4 version\_id**

The `version_id` specifies versioning information for the `Cable_pull_information`.

The `version_id` need not be specified for a particular `Cable_pull_information`.

#### **4.2.34 Cartesian\_coordinate\_space\_2d**

A Cartesian\_coordinate\_space\_2d is defined by two mutually perpendicular axes.

The data associated with a Cartesian\_coordinate\_space\_2d are the following:

- length\_measure\_unit;
- plane\_angle\_measure\_unit;
- precision.

##### **4.2.34.1 length\_measure\_unit**

The length\_measure\_unit specifies the increments used to define linear distances or sizes within a Cartesian\_coordinate\_space\_2d.

##### **4.2.34.2 plane\_angle\_measure\_unit**

The plane\_angle\_measure\_unit specifies the increments used to define angular distances within a Cartesian\_coordinate\_space\_2d.

##### **4.2.34.3 precision**

The precision specifies the numerical precision required for an appropriate handling of the items located in the Cartesian\_coordinate\_space\_2d.

See 4.3.312 for the application assertion.

#### **4.2.35 Cartesian\_coordinate\_space\_3d**

A Cartesian\_coordinate\_space\_3d describes a three-dimensional cartesian space.

The data associated with a Cartesian\_coordinate\_space\_3d are the following:

- length\_measure\_unit;
- plane\_angle\_measure\_unit;
- precision.

##### **4.2.35.1 length\_measure\_unit**

The length\_measure\_unit specifies the increments used to define linear distances or sizes within a Cartesian\_coordinate\_space\_3d.

##### **4.2.35.2 plane\_angle\_measure\_unit**

The plane\_angle\_measure\_unit specifies the increments used to define angular distances within a Cartesian\_coordinate\_space\_3d.

### 4.2.35.3 precision

The precision specifies the numerical precision required for an appropriate handling of the items located in the Cartesian\_coordinate\_space\_3d.

See 4.3.313 for the application assertion.

### 4.2.36 Cartesian\_coordinate\_space\_with\_grid

A Cartesian\_coordinate\_space\_with\_grid is a type of Cartesian\_coordinate\_space\_2d (see 4.2.34) that is a framework of spaced parallel lines with a spacing delta x and delta y used to ease the positioning of any graphical item.

The data associated with a Cartesian\_coordinate\_space\_with\_grid are the following:

- delta\_x;
- delta\_y;
- origin.

#### 4.2.36.1 delta\_x

The delta\_x specifies the spacing parallel to the x-axis.

#### 4.2.36.2 delta\_y

The delta\_y specifies the spacing parallel to the y-axis.

#### 4.2.36.3 origin

The origin specifies the positioning of the zero point of the Cartesian\_coordinate\_space\_with\_grid relative to the zero point of the Cartesian\_coordinate\_space\_2d (see 4.2.34).

### 4.2.37 Cartesian\_point

A Cartesian\_point is a position in space defined by its coordinates in a cartesian coordinate system.

The data associated with a Cartesian\_point are the following:

- coordinates.

#### 4.2.37.1 coordinates

The coordinates specifies the position of the Cartesian\_point in its coordinate space. The sequence of the coordinates in the list corresponds to the sequence of the coordinates as it is given by the mathematical definition of the coordinate system.

There shall be three or more coordinates for a Cartesian\_point.

### 4.2.38 Certification

## ISO 10303-212:2001(E)

A Certification is a certificate for an object.

The data associated with a Certification are the following:

- certification\_type;
- is\_applied\_to;
- name;
- purpose.

### 4.2.38.1 certification\_type

The certification\_type specifies the kind of certification.

EXAMPLE 'supplier certificate' is an example for the certification\_type.

### 4.2.38.2 is\_applied\_to

The is\_applied\_to specifies the object that the certificate is applied to.

Each is\_applied\_to may be one of the following: Device (see 4.2.88), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), or Supplier\_solution (see 4.2.334).

See 4.3.314, 4.3.315, 4.3.316, and 4.3.317 for the application assertions.

### 4.2.38.3 name

The name specifies the word or group of words by which the Certification is referred to.

### 4.2.38.4 purpose

The purpose specifies the objective of the Certification.

The purpose need not be specified for a particular Certification.

## 4.2.39 Chained\_dimension\_pair

A Chained\_dimension\_pair is a type of Dimension\_sequence\_pair (see 4.2.97) that is the relationship between two dimensions in which the terminus of one dimension initializes the next dimension in the sequence.

## 4.2.40 Class\_category\_association

A Class\_category\_association is the assignment of a Specification\_category (see 4.2.324) to a Product\_class (see 4.2.263). Additionally, this assignment specifies if the usage of one or more Specification (see 4.2.323) objects belonging to this Specification\_category (see 4.2.324) is mandatory or optional for all products of that Product\_class (see 4.2.263).



NOTE 1 In the case of a strict family concept for Specification\_category (see 4.2.324) objects, this Class\_category\_association specifies that the usage of one Specification (see 4.2.323) of this Specification\_category (see 4.2.324) is mandatory for all products of that Product\_class (see 4.2.263). If no strict family concept for Specification\_category (see 4.2.324) objects is used, this Class\_category\_association specifies that the usage of one or more Specification (see 4.2.323) objects of this Specification\_category (see 4.2.324) is mandatory or optional for products of that Product\_class (see 4.2.263).

NOTE 2 The assignment of a Specification\_category (see 4.2.324) to a Product\_class (see 4.2.263) cannot replace the association of single members of the Specification\_category (see 4.2.324) to the Product\_class (see 4.2.263).

The data associated with an Class\_category\_association are the following:

- associated\_category;
- associated\_product\_class;
- mandatory.

#### **4.2.40.1 associated\_category**

The associated\_category specifies the Specification\_category (see 4.2.324) that is associated with the Product\_class (see 4.2.263).

See 4.3.319 for the application assertion.

#### **4.2.40.2 associated\_product\_class**

The associated\_product\_class specifies the Product\_class (see 4.2.263) for which the Specification\_category (see 4.2.324) is valid.

See 4.3.318 for the application assertion.

#### **4.2.40.3 mandatory**

The mandatory specifies whether or not one or more Specification (see 4.2.323) objects have to be used for products within the referenced Product\_class (see 4.2.263). Nonmandatory Specification (see 4.2.323) objects are considered to be optional.

EXAMPLE The specification category 'radio' may be associated optional to the product class of a car; the specification category 'engine' is an example for a mandatory association.

#### **4.2.41 Class\_condition\_association**

A Class\_condition\_association is the relation between a Specification\_expression (see 4.2.326) and a Product\_class (see 4.2.263). This relationship contains the information that a particular Specification\_expression (see 4.2.326) is valid for all products of that product class.

## ISO 10303-212:2001(E)

The data associated with a Class\_condition\_association are the following:

- associated\_condition;
- associated\_product\_class;
- condition\_type;
- description.

### 4.2.41.1 associated\_condition

The associated\_condition specifies the Specification\_expression (see 4.2.326) that is assigned to the Product\_class (see 4.2.263).

See 4.3.321 for the application assertion.

### 4.2.41.2 associated\_product\_class

The associated\_product\_class specifies the Product class for which the Specification\_expression (see 4.2.326) is valid.

See 4.3.320 for the application assertion.

### 4.2.41.3 condition\_type

The condition\_type specifies the meaning of the association. The value is either user defined or predefined.

The predefined value of condition\_type is one of the following:

- design case;
- identification;
- part usage;
- validity.

NOTE See 4.2.41.3.1 - 4.2.41.3.3 for the definition of each predefined value for condition\_type.

#### 4.2.41.3.1 design case

design case: The Specification\_expression (see 4.2.326) specifies a condition when a given object has to be designed and verified. This value of the condition\_type is for information only and shall not be interpreted when querying design cases or usage cases. For such a query, the value of the attribute 'configuration\_type' of Configuration (see 4.2.56) shall be evaluated;

NOTE This value may be used to precise when a given Functionality (see 4.2.155) or a given Product\_component (see 4.2.265) has to be studied by the design department so that it provides solutions appropriate for the case specified by 'associated\_condition' and 'associated\_product\_class'.

#### 4.2.41.3.2 identification

identification: The Specification\_expression (see 4.2.326) specifies a condition that enables one to distinguish the associated Product\_class (see 4.2.263) from other Product\_class (see 4.2.263) objects. For a top-level node in a hierarchy of Product\_class (see 4.2.263) objects, this value is not applicable. This identification is part of the identification of all subclasses of this product class;

#### 4.2.41.3.3 part usage

part usage: The Specification\_expression (see 4.2.326) specifies a condition for the usage of the components of an Alternative\_solution (see 4.2.12) in the products of the associated Product\_class (see 4.2.263). In this case, the Class\_condition\_association shall be referenced by at least one Configuration (see 4.2.56) object;

#### 4.2.41.3.4 validity

validity: The Specification\_expression (see 4.2.326) specifies a condition that is used to verify a Product\_specification (see 4.2.269) for the associated Product\_class (see 4.2.263). That means that the Specification\_expression (see 4.2.326) evaluates to 'true' if the set of Specification (see 4.2.323) objects is valid; otherwise it evaluates to 'false' with the meaning that the specified object is invalid for the Product\_class (see 4.2.263). It is valid for all products belonging to the 'associated\_product\_class' in case of the condition types 'identification' and 'validity'.

#### 4.2.41.4 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Class\_condition\_association.

The description need not be specified for a particular Class\_condition\_association.

#### 4.2.42 Class\_inclusion\_association

A Class\_inclusion\_association is the assignment of a Specification\_inclusion (see 4.2.327) to a Product\_class (see 4.2.263). This assignment contains the information that a particular Specification\_inclusion (see 4.2.327) applies for all products of that Product\_class (see 4.2.263).

The data associated with a Class\_inclusion\_association are the following:

## ISO 10303-212:2001(E)

- associated\_inclusion;
- associated\_product\_class;
- description.

### 4.2.42.1 associated\_inclusion

The associated\_inclusion specifies the Specification\_inclusion (see 4.2.327) that is associated with the Product\_class (see 4.2.263).

See 4.3.323 for the application assertion.

### 4.2.42.2 associated\_product\_class

The associated\_product\_class specifies the Product\_class (see 4.2.263) for which the Specification\_inclusion (see 4.2.327) is valid.

See 4.3.322 for the application assertion.

### 4.2.42.3 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Class\_inclusion\_association.

The description need not be specified for a particular Class\_inclusion\_association.

### 4.2.43 Class\_reference

A Class\_reference specifies the information that is required to retrieve a General\_classification (see 4.2.156) object from a library compliant to ISO 13584-42.

NOTE 1 The library shall be compliant to the type referred within the 'type\_of\_repository' attribute.

NOTE 2 The library is not necessarily available in computer interpretable form.

EXAMPLE 1 A library available in paper form.

The data associated with a Class\_reference are the following:

- code;
- supplier;
- type\_of\_repository;
- version.

#### 4.2.43.1 code

The code specifies an identifier for the General\_classification (see 4.2.156) object within the repository. The format of this code is defined in ISO 13584-42.

NOTE 3 The repository referred within the 'type\_of\_repository' attribute may impose further restrictions on the content of this attribute.

#### 4.2.43.2 supplier

The supplier specifies the source of the repository. The format of this specification is defined in ISO 13584-26.

NOTE 4 The repository referred within the 'type\_of\_repository' attribute may impose further restrictions on the content of this attribute.

#### 4.2.43.3 type\_of\_repository

The type\_of\_repository specifies the kind of repository. The content of the attributes of Class\_reference and, if present, the associated Property\_reference (see 4.2.275) object shall follow the syntax specified by type\_of\_repository. The value is either user defined or predefined.

NOTE 5 In case when the attribute value is user defined the library shall still conform to ISO 13584.

EXAMPLE A company specific library 'RETSUA' that conforms to ISO 13584.

The predefined value of type\_of\_repository is one of the following:

- iec 61360 library;
- iso 13584 library.

NOTE See 4.2.43.3.1 - 4.2.43.3.2 for the definition of each predefined value for type\_of\_repository.

##### 4.2.43.3.1 iec 61360 library

iec 61360 library: The repository shall conform to IEC 61360.

NOTE 6 Libraries that conform to IEC 61360 add further restrictions to ISO 13584.

##### 4.2.43.3.2 iso 13584 library

## ISO 10303-212:2001(E)

iso 13584 library: The repository shall conform to ISO 13584.

### 4.2.43.1 version

The version specifies the variant of the entry in the repository.

NOTE 7 The repository referred within the 'type\_of\_repository' attribute may impose certain restrictions on the content of this attribute.

### 4.2.44 Class\_specification\_association

A Class\_specification\_association is the relation between a Specification (see 4.2.323) and a Product\_class (see 4.2.263). This Specification (see 4.2.323) serves as a potential characteristic of all products belonging to the product class.

The data associated with a Class\_specification\_association are the following:

- associated\_product\_class;
- associated\_specification;
- association\_type.

#### 4.2.44.1 associated\_product\_class

The associated\_product\_class specifies the Product\_class (see 4.2.263) for which the Specification (see 4.2.323) is valid.

See 4.3.324 for the application assertion.

#### 4.2.44.2 associated\_specification

The associated\_specification specifies the Specification (see 4.2.323) that is associated with the Product\_class (see 4.2.263).

See 4.3.325 for the application assertion.

#### 4.2.44.3 association\_type

The association\_type specifies the kind of availability of a particular Specification (see 4.2.323) in a Product\_class (see 4.2.263). The value is either user defined or predefined.

The predefined value of `association_type` is one of the following:

- availability;
- identification;
- non replaceable standard;
- part usage;
- option;
- replaceable standard.

NOTE See 4.2.44.3.1 - 4.2.44.3.5 for the definition of each predefined value for `association_type`.

#### **4.2.44.3.1 availability**

availability: The Specification (see 4.2.323) is a potential characteristic of any product belonging to a high-level Product class. It is not specified if this is an option or a standard.

#### **4.2.44.3.2 identification**

identification: The Specification (see 4.2.323) is the characteristic that enables to distinguish the associated Product\_class (see 4.2.263) from other Product\_class (see 4.2.263) objects. This is a kind of 'non replaceable standard'. For a top-level node in a hierarchy of Product\_class (see 4.2.263) objects this value shall not be applied. This identification is part of the identification of all subclasses of this Product\_class (see 4.2.263).

#### **4.2.44.3.3 non replaceable standard**

non replaceable standard: The Specification (see 4.2.323) is a characteristic of any product belonging to the product class.

EXAMPLE The specification 'South East Asia climate zone' is a 'non replaceable standard' for a system with market context 'Japan'.

#### **4.2.44.3.4 part usage**

part usage: The Specification (see 4.2.323) is a characteristic for the usage of the components of an Alternative\_solution (see 4.2.12) or the usage of an Device (see 4.2.88) in the products of the associated Product\_class (see 4.2.263).

#### **4.2.44.3.5 option**

option: The Specification (see 4.2.323) is a characteristic of a product if explicitly chosen. The Specification (see 4.2.323) replaces another Specification (see 4.2.323) of the same specification category if the replaced Specification (see 4.2.323) is associated with the Product class as 'replaceable standard'.

#### 4.2.44.3.6 replaceable standard

replaceable standard: The Specification (see 4.2.323) is a default characteristic of the products belonging to the Product\_class (see 4.2.263) as long as no other specification of the same specification category is chosen;

### 4.2.45 Class\_structure\_relationship

A Class\_structure\_relationship is the mechanism to state that the related application object is a means for fulfilling partially or fully the requirements of the relating Product\_class (see 4.2.263) or that the related application object is an element of the functional structure of the relating Product\_class (see 4.2.263).

The data associated with a Class\_structure\_relationship are the following:

- description;
- related;
- relating;
- relation type.

#### 4.2.45.1 description

The description specifies additional information about the Class\_structure\_relationship.

The description need not be specified for a particular Class\_structure\_relationship.

#### 4.2.45.2 related

The related specifies the Product\_component (see 4.2.265) or Single\_function\_unit (see 4.2.315) object related by the Class\_structure\_relationship.

Each related may be one of the following: Function\_definition (see 4.2.145) or Product\_component (see 4.2.265).

See 4.3.326 and 4.3.328 for the application assertions.

#### 4.2.45.3 relating

The relating specifies the Product\_class (see 4.2.263) object related by the Class\_structure\_relationship.

See 4.3.327 for the application assertion.

#### 4.2.45.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.



The predefined value of `relation_type` is one of the following:

- functionality;
- realization.

NOTE See 4.2.45.4.1 - 4.2.45.4.2 for the definition of each permissible value for `relation_type`.

#### 4.2.45.4.1 functionality

`functionality`: The related `Single_function_unit` (see 4.2.315) is an element of the functional structure of the relating `Product_class` (see 4.2.263).

#### 4.2.45.4.2 realization

`realization`: The related `Product_component` (see 4.2.265) is a means for fulfilling, partially or fully, the requirements identified with the relating `Product_class` (see 4.2.263).

### 4.2.46 Classification\_association

The `Classification_association` is the relation between the classification information and the item that is to be categorized. The purpose of this relationship is specified by the content of the attribute role.

The data associated with a `Classification_association` are the following:

- classification;
- `classified_element`;
- definitional;
- role.

#### 4.2.46.1 classification

The classification specifies the `General_classification` (see 4.2.156).

See 4.3.352 for the application assertion.

#### 4.2.46.2 classified\_element

The `classified_element` specifies the item that is categorized through the related `General_classification` (see 4.2.156).

Each `classified_element` may be one of the following: `Activity` (see 4.2.1), `Activity_method` (see 4.2.3), `Annotation_subfigure_definition` (see 4.2.18), `Approval` (see 4.2.23), `Approval_status` (see 4.2.25), `Complex_product` (see 4.2.51), `Connectivity_definition` (see 4.2.61), `Contract` (see 4.2.63), `Data_element` (see 4.2.70), `Data_element_association` (see 4.2.71), `Data_element_definition` (see 4.2.72), `Design_discipline_item_definition` (see 4.2.86), `Device` (see 4.2.88), `Document` (see 4.2.101), `Document_file` (see 4.2.106), `Document_version` (see 4.2.114), `Drawing` (see 4.2.119), `Drawing_sheet` (see 4.2.122), `Function_definition` (see 4.2.145), `Function_unit` (see 4.2.148), `Function_version` (see 4.2.150), `Functional_connectivity_definition` (see 4.2.152), `Functionality` (see 4.2.155), `Interface_port` (see 4.2.171), `Interface_terminal` (see 4.2.174), `Item` (see 4.2.178), `Item_version` (see

## ISO 10303-212:2001(E)

4.2.182), Location (see 4.2.192), Path (see 4.2.232), Path\_node (see 4.2.233), Port (see 4.2.247), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Project (see 4.2.271), Requirement (see 4.2.285), Route (see 4.2.290), Section (see 4.2.296), Section\_interface (see 4.2.298), Security\_level (see 4.2.302), Signal (see 4.2.309), Specification\_category (see 4.2.324), Terminal (see 4.2.338), Typical\_schematic\_node (see 4.2.347), User\_defined\_symbol\_definition (see 4.2.355), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.329, 4.3.330, 4.3.331, 4.3.332, 4.3.333, 4.3.334, 4.3.335, 4.3.336, 4.3.337, 4.3.338, 4.3.339, 4.3.340, 4.3.341, 4.3.342, 4.3.343, 4.3.344, 4.3.345, 4.3.346, 4.3.347, 4.3.348, 4.3.349, 4.3.350, 4.3.351, 4.3.353, 4.3.354, 4.3.355, 4.3.356, 4.3.357, 4.3.358, 4.3.359, 4.3.360, 4.3.361, 4.3.362, 4.3.363, 4.3.364, 4.3.365, 4.3.366, 4.3.367, 4.3.368, 4.3.369, 4.3.370, 4.3.371, 4.3.372, 4.3.373, 4.3.374, and 4.3.375 for the application assertions.

### 4.2.46.3 definitional

The definitional specifies whether or not the General\_classification (see 4.2.156) acts as an identifying characteristic of the item assigned by associated\_classification.

NOTE If definitional is 'true', the General\_classification (see 4.2.156) serves as an identification criterion for the associated item.

### 4.2.46.4 role

The role specifies the purpose of the General\_classification (see 4.2.156). The value is either user defined or predefined.

The role need not be specified for a particular Classification\_association. The role need not be specified for a particular Classification\_association.

The predefined value of role is one of the following:

- electromagnetic compatibility;
- environmental conditions;
- flow direction;
- protection class.

NOTE See 4.2.46.4.1 - 4.2.46.4.4 for the definition of each predefined value for role.

#### 4.2.46.4.1 electromagnetic compatibility

electromagnetic compatibility: The associated\_classification is the classification that categorizes the classified element in respect of its ability to comply with requirements concerning electromagnetic interference.

**4.2.46.4.2 environmental conditions**

environmental conditions: The associated\_classification is the classification that categorizes the classified element in respect of its ability to comply with requirements concerning its environmental surroundings.

NOTE Environmental conditions are classified in accordance to classification systems such as IEC 60721. Nonclassified data, such as the precise value of temperature, may be specified by using a Data\_element (see 4.2.70).

**4.2.46.4.3 flow direction**

flow direction: The associated\_classification is the classification that categorizes the classified element with respect to the flow of matter, energy, or information.

EXAMPLE 1 The assigned classification may give information whether the element acts as an input port, an output port, or a bi-directional port, etc.

**4.2.46.4.4 protection class**

protection class: The associated\_classification is the classification that categorizes the classified element in respect of its ability to withstand environmental conditions. The protection of the environment from potentially hazardous circumstances within a piece of equipment can be also addressed by this classification.

EXAMPLE 2 Protection from electrical shock, vibration, humidity, etc.

EXAMPLE 3 IEC 60529 specifies a classification for the degree of protection provided by enclosures.

NOTE To specify the protection class that is requested in a specific environment a Requirement (see 4.2.285) may be associated to the classified element.

**4.2.47 Classification\_attribute**

A Classification\_attribute is a characteristic used to classify an item associated with the corresponding General\_classification (see 4.2.156).

## ISO 10303-212:2001(E)

The data associated with a Classification\_attribute are the following:

- allowed\_value;
- associated\_classification;
- attribute\_definition;
- description;
- id;
- name.

### 4.2.47.1 allowed\_value

The allowed\_value specifies the set of Data\_element (see 4.2.70) objects that represent characteristic values of the Classification\_attribute.

See 4.3.376 for the application assertion.

### 4.2.47.2 associated\_classification

The associated\_classification specifies the General\_classification (see 4.2.156) the Classification\_attribute is a characteristic of.

See 4.3.378 for the application assertion.

### 4.2.47.3 attribute\_definition

The attribute\_definition specifies the Data\_element\_definition (see 4.2.72) that characterizes the allowed values.

NOTE The specification of compound characteristics can be realized by using Data\_element\_relationship (see 4.2.74) with relation\_type 'peer' or 'decomposition'.

See 4.3.377 for the application assertion.

### 4.2.47.4 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Classification\_attribute.

The description need not be specified for a particular Classification\_attribute.

### 4.2.47.5 id

The id specifies the identifier of the Classification\_attribute.

EXAMPLE The names 'a' or 'b' for length or width attributes or 'r' for radius attributes are examples for identifiers of Classification\_attribute objects. The meaning of such ids is usually specified in external sources.

#### 4.2.47.6 name

The name specifies a speaking designation of the Classification\_attribute.

The name need not be specified for a particular Classification\_attribute.

#### 4.2.48 Classification\_system

A Classification\_system is the scheme used to define the categorization of an item.

The data associated with a Classification\_system are the following:

- description;
- id.

##### 4.2.48.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Classification\_system.

The description need not be specified for a particular Classification\_system.

##### 4.2.48.2 id

The id specifies the identifier of the Classification\_system.

#### 4.2.49 Coded\_size

A Coded\_size is a type of Rectangular\_size (see 4.2.282) that is the specification of the paper size of a sheet in an abbreviated form.

EXAMPLE Examples for Coded\_size information are 'A4' or 'B5'.

The data associated with a Coded\_size are the following:

- size;
- referenced\_standard.

##### 4.2.49.1 size

The size specifies the information for the drawing sheet limits.

## ISO 10303-212:2001(E)

NOTE If the largeness differs from the dimensions specified by the Rectangular\_area (see 4.2.281) object, which is associated through Rectangular\_size (see 4.2.282), the dimensions specified through Rectangular\_area (see 4.2.281) apply.

### 4.2.49.2 referenced\_standard

The referenced\_standard specifies the normative basis that gives information how to interpret the size information.

See 4.3.379 for the application assertion.

### 4.2.50 Colour

A Colour is a characteristic of visual presentation that is the defined relationship between red, green, and blue proportions.

Each Colour is either a Predefined\_colour (see 4.2.251) or a User\_defined\_colour (see 4.2.350).

### 4.2.51 Complex\_product

A Complex\_product is an object with the capability that it can be realized by, decomposed into or specialized as Product\_constituent (see 4.2.266) objects in a functional, logical, or physical way. Each Complex\_product stands for a specific version. The capability of providing different views of the same Complex\_product or a version thereof is not supported.

Each Complex\_product is either an Alternative\_solution (see 4.2.12) or a Product\_component (see 4.2.265).

The data associated with a Complex\_product are the following:

- id;
- version\_id.

#### 4.2.51.1 id

The id specifies the identifier of the Complex\_product.

#### 4.2.51.2 version\_id

The version\_id specifies the identification of a particular version of a Complex\_product.

The version\_id need not be specified for a particular Complex\_product.

### 4.2.52 Complex\_product\_relationship

A Complex\_product\_relationship is a relationship between two Complex\_product (see 4.2.51) objects. The Complex\_product\_relationship shall only be used to relate Complex\_product (see 4.2.51) objects that are of the same kind.

The data associated with a Complex\_product\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### 4.2.52.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Complex\_product\_relationship.

The description need not be specified for a particular Complex\_product\_relationship.

#### 4.2.52.2 related

The related specifies the second of the two objects related by the Complex\_product\_relationship.

NOTE The semantics of this attribute are defined by the attribute relation\_type.

See 4.3.380 for the application assertion.

#### 4.2.52.3 relating

The relating specifies the first of the two objects related by the Complex\_product\_relationship.

NOTE The semantics of this attribute are defined by the attribute relation\_type.

See 4.3.381 for the application assertion.

#### 4.2.52.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- derivation;
- substitution;
- version hierarchy;
- version sequence.

NOTE 1 See 4.2.52.4.1 - 4.2.52.4.4 for the definition of each predefined value for relation\_type.

#### 4.2.52.4.1 derivation

derivation: The `Complex_product_relationship` defines a relationship where the related `Complex_product` (see 4.2.51) is derived from the relating `Complex_product` (see 4.2.51);

NOTE 2 The relationship does not imply inheritance of any kind between the application objects that are related.

#### 4.2.52.4.2 redundancy

redundancy: The `Complex_product_relationship` defines a relationship where the related `Complex_product` (see 4.2.51) is replicated by the relating `Complex_product` (see 4.2.51).

EXAMPLE 1 To provide for a fail-safe service a `Product_component` (see 4.2.265) is replicated. If one `Product_component` (see 4.2.265) fails, the other is still in service.

#### 4.2.52.4.3 substitution

substitution: The `Complex_product_relationship` defines a relationship where the related `Complex_product` (see 4.2.51) replaces the relating `Complex_product` (see 4.2.51).

#### 4.2.52.4.4 version hierarchy

version hierarchy: The `Complex_product_relationship` defines a relationship where the related `Complex_product` (see 4.2.51) is a sub version of the relating `Complex_product` (see 4.2.51);

EXAMPLE 2 'version hierarchy' is used, whenever a revision of one particular version is prepared, e.g., 'version 1.1'.

#### 4.2.52.4.5 version sequence

version sequence: The `Complex_product_relationship` defines a relationship where the relating `Complex_product` (see 4.2.51) is the preceding version and the related `Complex_product` (see 4.2.51) is the following version.

NOTE 3 The relationship does not imply inheritance of any kind between the application objects that are related.

EXAMPLE 3 'version sequence' is used, whenever a new version is prepared, e.g., 'version 1.0' is the preceding version for the following 'version 2.0'.

### 4.2.53 Component\_colour

A `Component_colour` is the sensation produced on the eye by rays of light reflected or emitted by the body of a piece of equipment.



The data associated with a Component\_colour are the following:

- coding\_system;
- colour\_id.

#### 4.2.53.1 coding\_system

The coding\_system specifies the method that shall be applied to interpret the content of the 'colour\_id' attribute.

The coding\_system need not be specified for a particular Component\_colour.

See 4.3.382 for the application assertion.

#### 4.2.53.2 colour\_id

The colour\_id specifies uniquely the identity of the colour coding\_system that is within scope of this part of ISO 10303.

### 4.2.54 Component\_placement

A Component\_placement is the information pertaining to the placement of a Product\_component (see 4.2.265), which is defined in its own coordinate space, in the coordinate space of a reference Product\_component (see 4.2.265).

**EXAMPLE** An example for the use of a Component\_placement is the placement of the Product\_component (see 4.2.265) 'motor control centre' dependent on requirements of the customer.

The data associated with a Component\_placement are the following:

- placed\_component;
- transformation;
- reference\_product\_component.

#### 4.2.54.1 placed\_component

The placed\_component applies only to objects that are unambiguously placed in the context of the placed\_component.

**NOTE** The Component\_placement does not define a placement for all variants of the placed\_component in the context of 'reference\_product\_component'.

See 4.3.383 for the application assertion.

#### 4.2.54.2 transformation

The transformation specifies the placement of the placed Product\_component (see 4.2.265).

### **4.2.54.3 reference\_product\_component**

The reference\_product\_component specifies the high level Product\_component (see 4.2.265) that is defined in the reference coordinate space.

See 4.3.384 for the application assertion.

### **4.2.55 Composition\_relationship**

A Composition\_relationship is the relation that specifies that a Function\_definition (see 4.2.145) consists of Function\_unit (see 4.2.148) objects.

The data associated with a Composition\_relationship are the following:

- composed\_function;
- functional\_component.

#### **4.2.55.1 composed\_function**

The composed\_function specifies a Function\_definition (see 4.2.145) that is composed of Function\_unit (see 4.2.148) objects.

See 4.3.385 for the application assertion.

#### **4.2.55.2 functional\_component**

The functional\_component specifies the Function\_unit (see 4.2.148) that is a constituent of the Function\_definition (see 4.2.145) object specified by composed\_function.

See 4.3.386 for the application assertion.

### **4.2.56 Configuration**

A Configuration is the association of a Class\_condition\_association (see 4.2.41) or a Class\_specification\_association (see 4.2.44) object with an application object in order to define a valid usage of this application object in the context of a certain Product\_class (see 4.2.263). The validity of the association may be limited by effectivity information.

**EXAMPLE** The validity of the association may be limited by a time period through assigning an Effectivity (see 4.2.127) object to it.

**NOTE** The semantics of the kind of association is defined by the attributes configuration\_type and inheritance\_type.

The data associated with a Configuration are the following:

- configuration\_type;
- configured\_element;
- inheritance\_type;
- is\_solution\_for.

### 4.2.56.1 configuration\_type

The configuration\_type specifies the valid usage of a Configuration object that is applied to the application object as configured element.

The value of configuration\_type is one of the following:

- design;
- usage.

**NOTE** See 4.2.56.1.1 - 4.2.56.1.2 for the definition of each permissible value for configuration\_type.

#### 4.2.56.1.1 design

**design:** The item referenced as configured element has to be designed and checked before it can actually be used in a given context. This context is specified by the Class\_condition\_association (see 4.2.41) and Class\_specification\_association (see 4.2.44) objects referenced as the is\_solution\_for.

#### 4.2.56.1.2 usage

**usage:** The item referenced as the configured element is controlled by a Configuration. The Class\_condition\_association (see 4.2.41) and Class\_specification\_association (see 4.2.44) objects specify the use cases and are referenced as the is\_solution\_for.

### 4.2.56.2 configured\_element

The configured\_element specifies the application object that is controlled for its valid usage by the Configuration.

Each configured\_element may be one of the following: Complex\_product (see 4.2.51), Connectivity\_definition (see 4.2.61), Device (see 4.2.88), Function\_unit (see 4.2.148), Functional\_connectivity\_definition (see 4.2.152), Location (see 4.2.192), or Signal (see 4.2.309).

See 4.3.389, 4.3.390, 4.3.391, 4.3.392, 4.3.393, 4.3.394, and 4.3.395 for the application assertions.

### 4.2.56.3 inheritance\_type

The inheritance\_type specifies whether or not an inheritance scheme for the configuration information in a hierarchical structure is applied to the application object referenced as the configured element. The levels within such a hierarchy are defined through Product\_structure\_-relationship (see 4.2.270) objects or the attribute 'base\_element' of Alternative\_solution (see 4.2.12) .

The value of inheritance\_type is one of the following:

- exception;
- inherited;
- local.

NOTE See 4.2.56.3.1 - 4.2.56.3.3 for the definition of each permissible value for inheritance\_type.

#### 4.2.56.3.1 exception

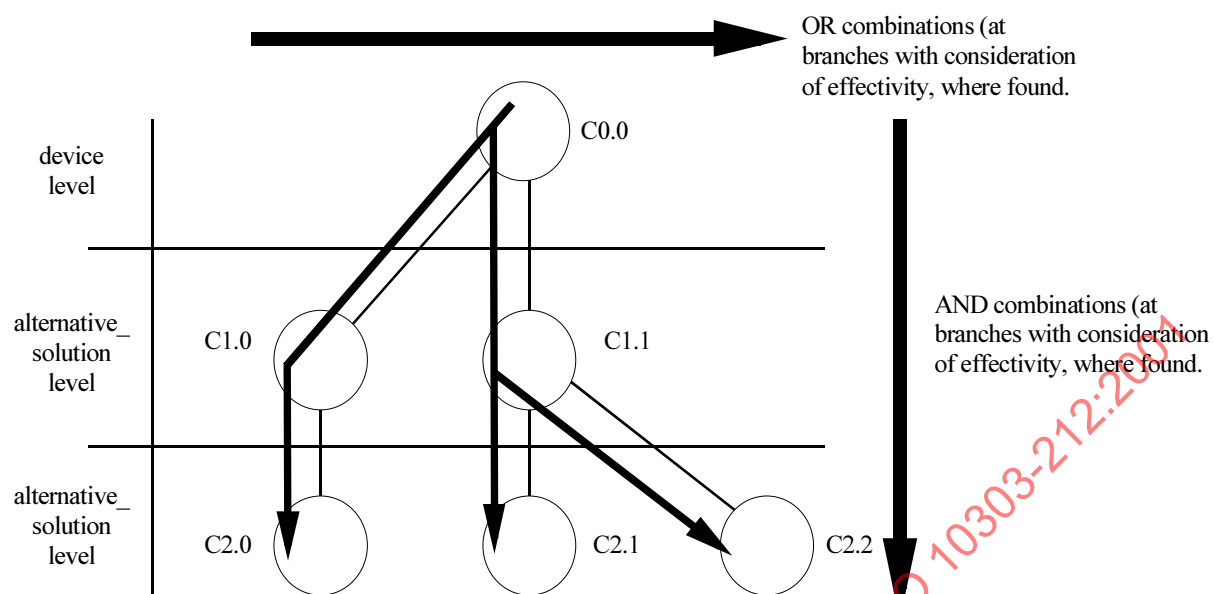
exception: No inheritance scheme is applicable and all required configuration information must be attached locally to the application object. The value indicates that the configuration information may be inconsistent to the structural levels above it or is on purpose contradictory to it. Therefore an inheritance scheme shall not continue beyond this point in the product structure tree.

EXAMPLE 1 A situation when the inheritance\_type 'exception' is applicable is a strike or some other disturbance unaccounted for in the current planning.

#### 4.2.56.3.2 inherited

inherited: 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 through AND combination of configuration information starting at the referenced configured element and OR combination for alternatives plus additional evaluation of effectivity information. The attribute value 'inherited' only applies for items for which the same value of configuration\_type is defined.

EXAMPLE 2 Figure 8 shows how inheritance is applied along the tree of a product structure: The complete configuration information of a Device (see 4.2.88) can be obtained by adding any such information to Alternative\_solution (see 4.2.12) objects which are linked through Solution\_instance\_assignment (see 4.2.318) objects and 'base\_element' attributes respectively. Whenever more than one higher level instance is present, the current information available is branched in as many branches as instances are present. For example, the total configuration information for the Device (see 4.2.88) given in Figure 8 could be expressed as follows: (C0.0 AND C1.0 AND C2.0) OR (C0.0 AND C1.1 AND C2.1) OR (C0.0 AND C1.1 AND C2.2).



**Figure 8 - Configuration inheritance scheme**

#### 4.2.56.3.3 local

local: No inheritance scheme is applicable and all required configuration information must be attached locally at the application object. Nevertheless, any potentially inherited configuration information of a higher level shall be consistent, i.e., be a subset of the locally defined configuration information;

#### 4.2.56.4 is\_solution\_for

The `is_solution_for` specifies the characteristics for which the item referenced as configured element provides a solution. The characteristics are described by a `Class_specification_association` (see 4.2.44). Combinations of characteristics are defined by a `Class_condition_association` (see 4.2.41) where the 'condition\_type' attribute is 'part usage'.

Each `is_solution_for` may be one of the following: `Class_condition_association` (see 4.2.41) or `Class_specification_association` (see 4.2.44).

See 4.3.387 and 4.3.388 for the application assertions.

#### 4.2.57 Connect\_area

A `Connect_area` is the specification of the zone where the schematic terminal is allowed to be connected with other terminals or connecting lines.

## ISO 10303-212:2001(E)

The data associated with a `Connect_area` are the following:

— `defined_by`.

### 4.2.57.1 `defined_by`

The `defined_by` specifies the borders of an area or a point that defines a zone where the schematic terminal is allowed to be connected with other terminals or connecting lines. That zone is sometimes addressed as the 'hot spot' of the connect node.

Each `defined_by` may be one of the following: `Curve_2d` (see 4.2.66) or `Point_2d` (see 4.2.245).

See 4.3.396 and 4.3.397 for the application assertions.

### 4.2.58 `Connecting_line`

A `Connecting_line` is the representation of a `Connection` (see 4.2.59) in a schematic diagram. It shall be used to present connectivity.

The data associated with a `Connecting_line` are the following:

— `presents`.

#### 4.2.58.1 `presents`

The `presents` specifies the connectivity items the `Connecting_line` is displaying.

Each `presents` may be one of the following: `Connectivity_definition` (see 4.2.61) or `Functional_connectivity_definition` (see 4.2.152).

See 4.3.398 and 4.3.399 for the application assertions.

### 4.2.59 `Connection`

A `Connection` is a type of `Connectivity_definition` (see 4.2.61) that is a link between `Terminal` (see 4.2.338) objects with the intention to allow the flow of information, energy, or matter. `Connection` objects may be used to divide a network in portions that can be implemented by using a specified equipment.

NOTE 1 The connections represent the information that is normally given in connection tables.

NOTE 2 The internal structure of connection bundles can be specified by decomposing a Connection into its constituting Connection objects by assigning Connectivity\_definition\_-relationship (see 4.2.62) objects of relation\_type 'decomposition'.

EXAMPLE 1 Implementing a network by using busbars may require another subdivision into Connection objects as if the network is to be implemented with cables.

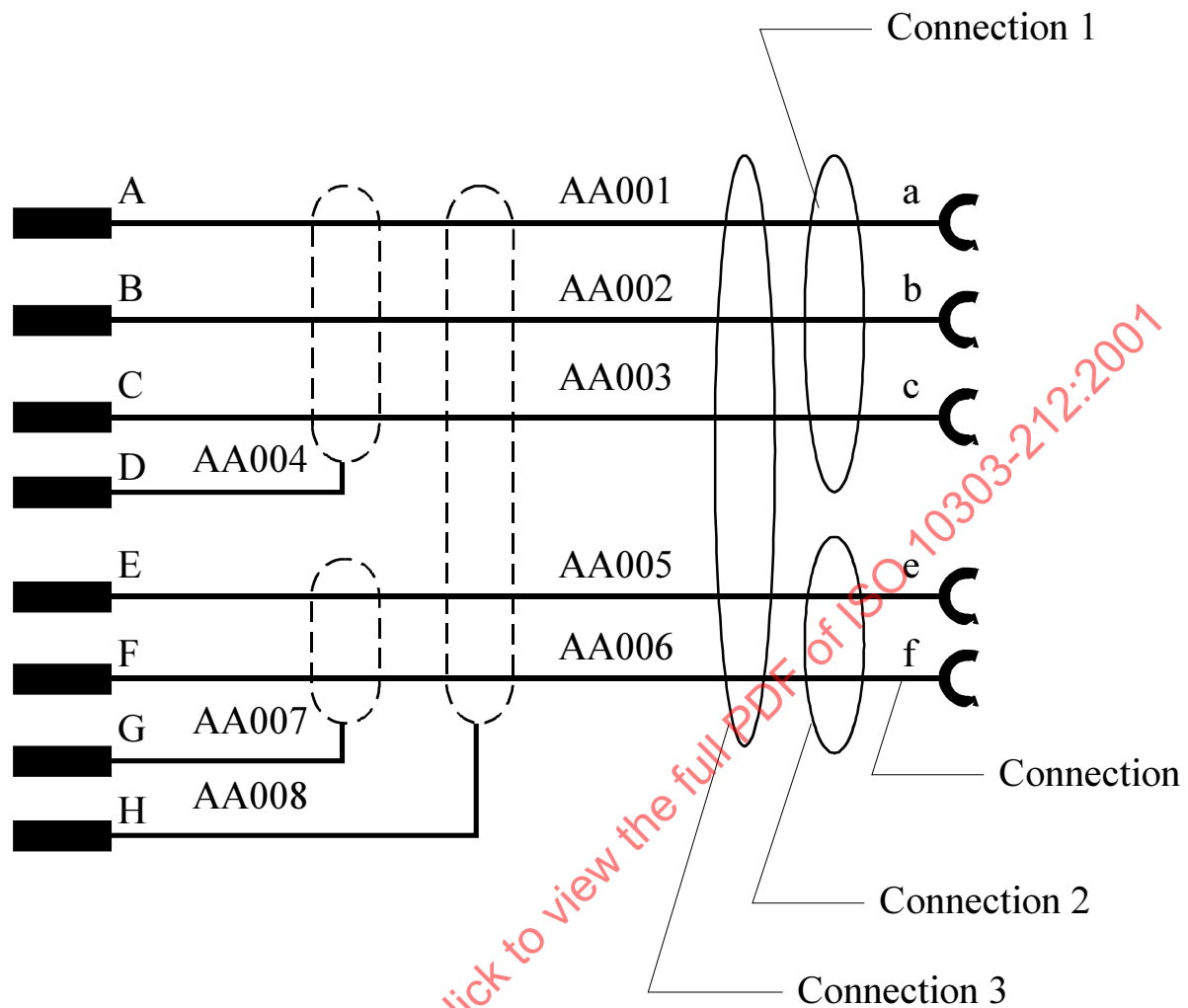
EXAMPLE 2 In Figure 9 a cable is shown. Each pin of the cable is defined as a Terminal (see 4.2.338), and each wire in the cable is defined as a Connection. In this figure the five Connection objects are grouped. Connection 3 decomposes into Connection objects 1 and 2. Connection 1 decomposes into three Connection objects, Connection 2 consists of two Connection objects.

The data associated with a connection are the following:

— connected\_terminal.

#### 4.2.59.1 connected\_terminal

The connected\_terminal specifies the terminals interconnected by the Connection.



**Figure 9 - Decomposition of Connection objects**

See 4.3.400 for the application assertion.

#### 4.2.60 Connectivity\_allocation

A Connectivity\_allocation is the relation that specifies the item selected to implement the specified Functional\_connectivity\_definition (see 4.2.152).



The data associated with a Connectivity\_allocation are the following:

- allocated\_connectivity\_definition;
- connectivity\_implementation;
- description.

#### **4.2.60.1 allocated\_connectivity\_definition**

The allocated\_connectivity\_definition specifies Functional\_connectivity\_definition (see 4.2.152) objects or Device (see 4.2.88) objects that are assigned by the Connectivity\_allocation to Connectivity\_definition (see 4.2.61) objects.

See 4.3.404 for the application assertion.

#### **4.2.60.2 connectivity\_implementation**

The connectivity\_implementation specifies Connectivity\_definition (see 4.2.61) elements that are assigned to Functional\_connectivity\_definition (see 4.2.152) elements or Device (see 4.2.88) objects.

Each connectivity\_implementation may be one of the following: Connectivity\_definition (see 4.2.61), Device (see 4.2.88), Function\_unit (see 4.2.148), or Physical\_instance (see 4.2.243).

See 4.3.401, 4.3.402, 4.3.403, and 4.3.405 for the application assertions.

#### **4.2.60.3 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Connectivity\_allocation.

The description need not be specified for a particular Connectivity\_allocation.

#### **4.2.61 Connectivity\_definition**

A Connectivity\_definition is the specification of the ability to enable a flow of information, energy, or material within a piece of equipment.

Each Connectivity\_definition is either a Connection (see 4.2.59) or an Interface\_terminal\_connection (see 4.2.175).

## ISO 10303-212:2001(E)

The data associated with a Connectivity\_definition are the following:

- connectivity\_of;
- description;
- id;
- implemented\_by;
- version\_id.

### 4.2.61.4 connectivity\_of

The connectivity\_of specifies the Assembly\_definition (see 4.2.27) object whose internal connectivity is specified by the Connectivity\_definition.

See 4.3.406 for the application assertion.

### 4.2.61.5 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Connectivity\_definition.

The description need not be specified for a particular Connectivity\_definition.

### 4.2.61.6 id

The id specifies the identifier of the Connectivity\_definition.

### 4.2.61.7 implemented\_by

The implemented\_by specifies the Device (see 4.2.88) objects that are used to accomplish the connectivity specified by Connectivity\_definition.

Each implemented\_by may be one of the following: Device (see 4.2.88) or Physical\_instance (see 4.2.243).

See 4.3.407 and 4.3.408 for the application assertions.

### 4.2.61.8 version\_id

The version\_id specifies versioning information for the Connectivity\_definition.

The version\_id need not be specified for a particular Connectivity\_definition.

## 4.2.62 Connectivity\_definition\_relationship

A Connectivity\_definition\_relationship is the relation between two Connectivity\_definition (see 4.2.61) objects.

The data associated with an Connectivity\_definition\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### **4.2.62.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Connectivity\_definition\_relationship.

The description need not be specified for a particular Connectivity\_definition\_relationship.

#### **4.2.62.2 related**

The related specifies the second of the two Connectivity\_definition (see 4.2.61) objects related by the Connectivity\_definition\_relationship.

See 4.3.409 for the application assertion.

#### **4.2.62.3 relating**

The relating specifies the first of the two Connectivity\_definition (see 4.2.61) objects related by the Connectivity\_definition\_relationship.

See 4.3.410 for the application assertion.

#### **4.2.62.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- alternate;
- decomposition;
- derivation;
- redundancy;
- substitution;
- version hierarchy;
- version sequence.

NOTE 1 See 4.2.62.4.1 - 4.2.62.4.7 for the definition of each predefined value for `relation_type`.

### 4.2.62.4.1 alternate

alternate: The `Connectivity_definition_relationship` defines a relationship where the related `Connectivity_definition` (see 4.2.61) is a possible substitute to the relating `Connectivity_definition` (see 4.2.61).

NOTE 2 This concept refers to the possibility to replace the related `Connectivity_definition` (see 4.2.61). The actual replacement is addressed by 'substitution'.

### 4.2.62.4.2 decomposition

decomposition: The `Connectivity_definition_relationship` defines a relationship where the related `Connectivity_definition` (see 4.2.61) is one of the components into which the relating `Connectivity_definition` (see 4.2.61) is divided up.

### 4.2.62.4.3 derivation

derivation: The `Connectivity_definition_relationship` defines a deriving relationship where the related `Connectivity_definition` (see 4.2.61) is based on the relating `Connectivity_definition` (see 4.2.61).

### 4.2.62.4.4 redundancy

redundancy: The `Connectivity_definition_relationship` defines a relationship where the related `Connectivity_definition` (see 4.2.61) is replicated by the relating `Connectivity_definition` (see 4.2.61).

EXAMPLE To provide for a fail-safe link two connections are used to provide for the connectivity. If one connection fails, then the other is still in service.

### 4.2.62.4.5 substitution

substitution: The `Connectivity_definition_relationship` defines a relationship where the related `Connectivity_definition` (see 4.2.61) replaces the relating `Connectivity_definition` (see 4.2.61).

### 4.2.62.4.6 version hierarchy

version hierarchy: The `Connectivity_definition_relationship` defines a hierarchical relationship where the related `Connectivity_definition` (see 4.2.61) is a subversion of the relating `Connectivity_definition` (see 4.2.61).

EXAMPLE Revisions 1.1 and 1.2 of a `Connectivity_definition` (see 4.2.61).

#### 4.2.62.4.7 version sequence

version sequence: The `Connectivity_definition_relationship` defines a succession of versions where the relating `Connectivity_definition` (see 4.2.61) is the preceding version, and the related `Connectivity_definition` (see 4.2.61) is the following version.

### 4.2.63 Contract

A Contract is the binding agreement that specifies the connection between the assigned parties.

The data associated with a Contract are the following:

- `contracted_element`;
- `description`;
- `id`.

#### 4.2.63.1 contracted\_element

The `contracted_element` specifies the work that is the subject of the Contract.

Each `contracted_element` may be one of the following: `Activity` (see 4.2.1), `Data_element` (see 4.2.70), `Design_discipline_item_definition` (see 4.2.86), `Device` (see 4.2.88), `Function_definition` (see 4.2.145), `Function_unit` (see 4.2.148), `Function_version` (see 4.2.150), `Item_version` (see 4.2.182), `Location` (see 4.2.192), `Node` (see 4.2.208), `Notification` (see 4.2.213), `Path` (see 4.2.232), `Path_node` (see 4.2.233), `Physical_instance` (see 4.2.243), `Process_variable` (see 4.2.260), `Project` (see 4.2.271), `Route` (see 4.2.290), `Section` (see 4.2.296), `Section_end` (see 4.2.297), `Section_interface` (see 4.2.298), `Signal` (see 4.2.309), `Signal_value` (see 4.2.313), `Technical_system` (see 4.2.336), `Work_order` (see 4.2.364), or `Work_request` (see 4.2.365).

See 4.3.411, 4.3.412, 4.3.413, 4.3.414, 4.3.415, 4.3.416, 4.3.417, 4.3.418, 4.3.419, 4.3.420, 4.3.421, 4.3.422, 4.3.423, 4.3.424, 4.3.425, 4.3.426, 4.3.427, 4.3.428, 4.3.429, 4.3.430, 4.3.431, 4.3.432, 4.3.433, 4.3.434, and 4.3.435 for the application assertions.

#### 4.2.63.2 description

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the Contract.

The `description` need not be specified for a particular Contract.

#### 4.2.63.3 id

The `id` specifies the identifier of the Contract.

#### **4.2.64 Cross\_reference**

A Cross\_reference is an element that provides the reference from one part of a document to another.

Each Cross\_reference is either a Detached\_representation\_reference (see 4.2.87), a Note\_reference (see 4.2.212), or a Page\_connector\_reference (see 4.2.229).

#### **4.2.65 Cross\_section**

A Cross\_section is the transverse section of a solid.

The data associated with a Cross\_section are the following:

— value\_of\_cross\_section.

##### **4.2.65.1 value\_of\_cross\_section**

The value\_of\_cross\_section specifies the numerical value that represents the extent of the cross section.

See 4.3.436 for the application assertion.

#### **4.2.66 Curve\_2d**

A Curve\_2d is the path of a point moving in a two-dimensional coordinate space.

#### **4.2.67 Curve\_3d**

A Curve\_3d is the path of a point moving in a three-dimensional coordinate space.

#### **4.2.68 Curve\_appearance**

A Curve\_appearance is a type of Appearance (see 4.2.21) that governs the visual presentation of geometric curves and annotation curves.

The data associated with a Curve\_appearance are the following:

- corner\_styles;
- curve\_colour;
- curve\_ends;
- draughting\_role;
- font;
- width.

#### 4.2.68.1 corner\_styles

The corner\_styles specifies whether the appearance of an Annotation\_curve (see 4.2.14) is squared or rounded at its corners.

The value of corner\_styles is one of the following:

- round;
- square.

NOTE See 4.2.68.1.1 - 4.2.68.1.2 for the definition of each permissible value for relation\_type.

##### 4.2.68.1.1 round

round: The graphical presentation of a corner with corner\_style round is shown in Figure 10.

##### 4.2.68.1.2 square

square: The graphical presentation of a corner in a line with corner\_style square is shown in Figure 10.



**squared corner**



**round corner**

**Figure 10 - Predefined corner styles**

The corner\_styles need not be specified for a particular Curve\_appearance.

#### **4.2.68.2 curve\_colour**

The curve\_colour specifies the colour to be used for the presentation of a curve.

See 4.3.437 for the application assertion.

#### **4.2.68.3 curve\_ends**

The curve\_ends specifies whether the appearance of an Annotation\_curve (see 4.2.14) is squared or rounded at its ends.

The value of curve\_ends is one of the following:

- round;
- square.

NOTE See 4.2.68.3.1 - 4.2.68.3.2 for the definition of each permissible value for curve\_ends.

##### **4.2.68.3.1 round**

round: The graphical presentation of a line with curve\_ends of type round is shown in Figure 11.

##### **4.2.68.3.2 square**

square: The graphical presentation of a line with curve\_ends of type square is shown in Figure 11.





**Figure 11 - Predefined curve end styles**

The curve\_ends need not be specified for a particular Curve\_appearance.

#### **4.2.68.4 draughting\_role**

The draughting\_role specifies the purpose within a draughting for a particular curve appearance.

EXAMPLE A draughting\_role could be a centreline or section line.

The draughting\_role need not be specified for a particular Curve\_appearance.

#### **4.2.68.5 font**

The font describes the style to be applied on the presentation of a curve.

See 4.3.438 for the application assertion.

#### **4.2.68.6 width**

The width specifies the thickness of the curve measured perpendicular to the direction of the curve.

### **4.2.69 Curve\_dimension**

A Curve\_dimension is a type of Dimension (see 4.2.93) that is the graphical presentation of the value of the distance between two elements, measured along a curved path or the length of a curved element.

The data associated with a Curve\_dimension are the following:

- component;
- extent.

#### **4.2.69.1 component**

The component specifies the projection lines that shows the extension of the points, lines, or surfaces that bound the measurement.

See 4.3.440 for the application assertion.

#### **4.2.69.2 extent**

The extent specifies the dimension line that graphically presents where the dimension value applies.

See 4.3.439 for the application assertion.

## **4.2.70 Data\_element**

A Data\_element is a technical or administrative property that may be used to characterize items or relationships between items. Data\_element objects allow one to associate technical or administrative data to the product data.

NOTE 1 Data associated to distinct life cycle stages shall be associated to application objects that are related to Function\_definition (see 4.2.145), Design\_discipline\_item\_definition (see 4.2.86), or Product\_component (see 4.2.265) objects with the appropriate Application\_context (see 4.2.22) object assigned.

Each Data\_element is either a Predefined\_data\_element (see 4.2.253) or a User\_defined\_data\_element (see 4.2.351).

The data associated with a Data\_element are the following:

- description;
- global\_unit;
- qualifier;
- value\_determination.

### **4.2.70.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Data\_element.

The description need not be specified for a particular Data\_element.

### **4.2.70.2 global\_unit**

The global\_unit specifies a unit that is valid for all Value\_with\_unit (see 4.2.360) objects that are referenced by the Data\_element.

The global\_unit need not be specified for a particular Data\_element.

### **4.2.70.3 qualifier**

The qualifier specifies the interpretation of the value assigned to Data\_element.

NOTE 2 The value is assigned through the 'value\_of\_data\_element' attribute of User\_defined\_data\_element (see 4.2.351) or by the 'data' attribute of Predefined\_data\_element (see 4.2.253).

The value of qualifier is one of the following:

- nominal;
- specified;
- typical.

NOTE 3 See 4.2.70.3.1 - 4.2.70.3.3 for the definition of each permissible value for qualifier.

#### **4.2.70.3.1 nominal**

nominal: The value assigned to Data\_element represents a suitable approximate quantity value used to designate the assigned item.

#### **4.2.70.3.2 specified**

specified: The value assigned to Data\_element represents the value as it is included in the specification. The actual value may differ from the specified value.

#### **4.2.70.3.3 typical**

typical: The value assigned to Data\_element represents a value that is characteristic for that specific attribute under the circumstances the attribute is used.

The qualifier need not be specified for a particular Data\_element.

#### **4.2.70.4 value\_determination**

The value\_determination specifies the kind of data given by Data\_element. The value is either user defined or predefined.

The value\_determination need not be specified for a particular Data\_element. The value\_determination need not be specified for a particular Data\_element.

## ISO 10303-212:2001(E)

The predefined value of value\_determination is one of the following:

- calculated;
- designed;
- estimated;
- measured;
- required;
- setpoint.

NOTE 4 See 4.2.70.4.1 - 4.2.70.4.6 for the definition of each predefined value for value\_determination.

### 4.2.70.4.1 **calculated**

calculated: The value has been calculated from a particular representation.

### 4.2.70.4.2 **designed**

designed: The value represents the value intended by the design.

### 4.2.70.4.3 **estimated**

estimated: The value has been estimated.

### 4.2.70.4.4 **measured**

measured: The value has been measured.

### 4.2.70.4.5 **required**

required: The value represents the requirement.

### 4.2.70.4.6 **setpoint**

setpoint: The value represents the quantity that is to be used as specified value.

## 4.2.71 **Data\_element\_association**

A Data\_element\_association is the assignment of a Data\_element (see 4.2.70) to an item.

The data associated with a Data\_element\_association are the following:

- associated\_data\_element;
- associated\_item;
- definitional;
- description;
- data\_element\_context.

#### 4.2.71.1 associated\_data\_element

The associated\_data\_element specifies the Data\_element (see 4.2.70) that is assigned to an item.

See 4.3.450 for the application assertion.

#### 4.2.71.2 associated\_item

The associated\_item specifies the item that is described by the associated Data\_element (see 4.2.70).

Each associated\_item may be one of the following: Activity (see 4.2.1), Assembly\_component\_relationship (see 4.2.26), Cable\_pull\_information (see 4.2.33), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Connectivity\_allocation (see 4.2.60), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Drawing\_sheet (see 4.2.122), Free\_segment (see 4.2.144), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), Functionality (see 4.2.155), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Offered\_function\_allocation (see 4.2.220), Path (see 4.2.232), Path\_node (see 4.2.233), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Port\_association (see 4.2.249), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Requirement (see 4.2.285), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Signal (see 4.2.309), Signal\_value (see 4.2.313), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), or Work\_order (see 4.2.364).

See 4.3.441, 4.3.442, 4.3.443, 4.3.444, 4.3.445, 4.3.446, 4.3.447, 4.3.448, 4.3.449, 4.3.451, 4.3.452, 4.3.453, 4.3.454, 4.3.455, 4.3.456, 4.3.457, 4.3.458, 4.3.459, 4.3.460, 4.3.461, 4.3.462, 4.3.463, 4.3.464, 4.3.465, 4.3.466, 4.3.467, 4.3.468, 4.3.469, 4.3.470, 4.3.471, 4.3.472, 4.3.473, 4.3.474, 4.3.475, 4.3.476, 4.3.477, 4.3.478, 4.3.479, 4.3.480, 4.3.481, 4.3.482, 4.3.483, 4.3.485, 4.3.486,

## ISO 10303-212:2001(E)

4.3.487, 4.3.488, 4.3.489, 4.3.490, 4.3.491, 4.3.492, 4.3.493, 4.3.494, 4.3.495, 4.3.497, 4.3.498, 4.3.499, 4.3.500, 4.3.501, 4.3.502, 4.3.503, 4.3.504, 4.3.505, 4.3.506, 4.3.507, 4.3.508, 4.3.509, 4.3.510, 4.3.512, 4.3.513, and 4.3.514 for the application assertions.

### 4.2.71.3 definitional

The definitional specifies whether or not the associated Data\_element (see 4.2.70) acts as an identifying characteristic of the item assigned by associated\_item.

### 4.2.71.4 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Data\_element\_association.

The description need not be specified for a particular Data\_element\_association.

### 4.2.71.5 data\_element\_context

The data\_element\_context specifies an environment for which the Data\_element\_association is valid.

The data\_element\_context need not be specified for a particular Data\_element\_association.

Each data\_element\_context may be one of the following: Organization (see 4.2.223), Product\_class (see 4.2.263), or Technical\_system (see 4.2.336).

See 4.3.484, 4.3.496, and 4.3.511 for the application assertions.

### 4.2.72 Data\_element\_definition

A Data\_element\_definition is the composition of the information that specifies the meaning of the Data\_element (see 4.2.70).

The data associated with an Data\_element\_definition are the following:

- admitted\_qualifier;
- allowed\_unit;
- description;
- id;
- source;
- version\_id.

#### 4.2.72.1 admitted\_qualifier

The admitted\_qualifier specifies the permitted level that the data element which is associated to the Data\_element\_definition may possess.

NOTE 1 If present, only the levels specified by the 'admitted\_qualifier' attribute may occur in the corresponding Data\_element (see 4.2.70) object.

The value of admitted\_qualifier is one of the following:

- nominal;
- specified;
- typical.

NOTE 2 See 4.2.72.1.1 - 4.2.72.1.5 for the definition of each permissible value for admitted\_qualifier.

#### **4.2.72.1.1 nominal**

nominal: The associated value represents a suitable approximate quantity used to designate an item.

#### **4.2.72.1.2 specified**

specified: The associated value represents the value as it is included in the specification. The actual value may differ from the specified value.

#### **4.2.72.1.3 typical**

typical: The associated value represents the value that is characteristic for that specific attribute under the circumstances the attribute is used.

### **4.2.72.2 allowed\_unit**

The allowed\_unit specifies the units the data element which is associated to the Data\_element\_definition may possess.

NOTE If present, only the units specified by the 'allowed\_unit' attribute may occur in the corresponding Data\_element (see 4.2.70) object.

There shall be zero, one or more allowed\_unit for a Data\_element\_definition.

### **4.2.72.3 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Data\_element\_definition.

The description need not be specified for a particular Data\_element\_definition.

### **4.2.72.4 id**

The id specifies the identifier of the Data\_element\_definition.

### **4.2.72.5 source**

The source specifies the query information to retrieve the Data\_element\_definition from a repository.

## **ISO 10303-212:2001(E)**

The source need not be specified for a particular Data\_element\_definition.

Each source may be one of the following: External\_library\_reference (see 4.2.132) or Property\_reference (see 4.2.275).

See 4.3.515 and 4.3.516 for the application assertions.

### **4.2.72.6 version\_id**

The version\_id specifies versioning information for the Data\_element\_definition.

The version\_id need not be specified for a particular Data\_element\_definition.

### **4.2.73 Data\_element\_definition\_relationship**

A Data\_element\_definition\_relationship is the relation between two Data\_element\_definition (see 4.2.72) objects.

The data associated with an Data\_element\_definition\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### **4.2.73.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Data\_element\_definition\_relationship.

The description need not be specified for a particular Data\_element\_definition\_relationship.

#### **4.2.73.2 related**

The related specifies the second of the two Data\_element\_definition (see 4.2.72) objects related by the Data\_element\_definition\_relationship.

See 4.3.517 for the application assertion.

#### **4.2.73.3 relating**

The relating specifies the first of the two Data\_element\_definition (see 4.2.72) objects related by the Data\_element\_definition\_relationship.

See 4.3.518 for the application assertion.



#### 4.2.73.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- decomposition;
- dependency;
- substitution;
- value domain.

NOTE See 4.2.73.4.1 - 4.2.73.4.4 for the definition of each predefined value for relation\_type.

##### 4.2.73.4.1 decomposition

decomposition: The Data\_element\_definition\_relationship defines a relationship where the related Data\_element\_definition (see 4.2.72) is member of a group of Data\_element\_definition (see 4.2.72) objects that is established by the relating Data\_element\_definition (see 4.2.72).

##### 4.2.73.4.2 dependency

dependency: The Data\_element\_definition\_relationship defines a relationship where the related Data\_element\_definition (see 4.2.72) is dependent upon the relating Data\_element\_definition (see 4.2.72).

##### 4.2.73.4.3 substitution

substitution: The Data\_element\_definition\_relationship defines a relationship where the related Data\_element\_definition (see 4.2.72) replaces the relating Data\_element\_definition (see 4.2.72).

##### 4.2.73.4.4 value domain

value domain: The Data\_element\_definition\_relationship defines a relationship where the values assigned to the related Data\_element\_definition (see 4.2.72) shall be within the limits indicated by the values assigned to the relating Data\_element\_definition (see 4.2.72).

EXAMPLE The output range of a 4-bit analog to digital converter is restricted to 16 discrete values, while the input voltage is restricted to a continuous range.

#### 4.2.74 Data\_element\_relationship

A Data\_element\_relationship is the relation between two Data\_element (see 4.2.70) objects.

## ISO 10303-212:2001(E)

The data associated with an Data\_element\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.74.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Data\_element\_relationship.

The description need not be specified for a particular Data\_element\_relationship.

### 4.2.74.2 related

The related specifies the second of the two Data\_element (see 4.2.70) objects related by the Data\_element\_relationship.

See 4.3.519 for the application assertion.

### 4.2.74.3 relating

The relating specifies the first of the two Data\_element (see 4.2.70) objects related by the Data\_element\_relationship.

See 4.3.520 for the application assertion.

### 4.2.74.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- decomposition;
- dependency;
- equivalence;
- substitution;
- tolerancing information.

NOTE 1 See 4.2.74.4.1 - 4.2.74.4.5 for the definition of each predefined value for relation\_type.

NOTE 2 Data\_element (see 4.2.70) objects that are related to each other in a sequential manner shall be captured as a one-dimensional array using Aggregated\_value (see 4.2.7) objects.

**4.2.74.4.1 decomposition**

decomposition: The Data\_element\_relationship defines a relationship where the related Data\_element (see 4.2.70) is one of the members of the group identified by the relating Data\_element (see 4.2.70).

**4.2.74.4.2 dependency**

dependency: The Data\_element\_relationship defines a deriving relationship where the related Data\_element (see 4.2.70) is based on the relating Data\_element (see 4.2.70).

**4.2.74.4.3 equivalence**

equivalence: The Data\_element\_relationship defines a relationship where the related Data\_element (see 4.2.70) represents the same matter of fact as by the relating Data\_element (see 4.2.70).

EXAMPLE Two length values shall be considered to be the same, regardless of rounding differences.

**4.2.74.4.4 substitution**

substitution: The Data\_element\_relationship defines a relationship where the related Data\_element (see 4.2.70) replaces the relating Data\_element (see 4.2.70).

**4.2.74.4.5 tolerancing information**

tolerancing information: The Data\_element\_relationship defines a relationship where the related Data\_element (see 4.2.70) provides information on the allowable variation of the value of the relating Data\_element (see 4.2.70).

**4.2.75 Data\_element\_specification**

The Data\_element\_specification is the defining description of a Data\_element\_definition (see 4.2.72).

The data associated with an Data\_element\_specification are the following:

- definition;
- language\_specification;
- note;
- remark;
- specification\_of.

**4.2.75.1 definition**

The definition specifies the meaning of the associated Data\_element (see 4.2.70) in human-interpretable language. It provides differentiation from all other Data\_element (see 4.2.70) objects.

#### **4.2.75.2 language\_specification**

The language\_specification specifies a language spoken by human beings to communicate with each other verbally or in written form.

The language\_specification need not be specified for a particular Data\_element\_specification.

See 4.3.522 for the application assertion.

#### **4.2.75.3 note**

The note specifies a statement that provides further information on the definition, which is essential to the understanding of the definition.

The note need not be specified for a particular Data\_element\_specification.

#### **4.2.75.4 remark**

The remark specifies human-interpretable text that gives further details about the Data\_element\_specification.

The remark need not be specified for a particular Data\_element\_specification.

#### **4.2.75.5 specification\_of**

The specification\_of specifies the Data\_element\_definition (see 4.2.72).

See 4.3.521 for the application assertion.

#### **4.2.76 Data\_element\_value**

A Data\_element\_value is the logical, numerical, or textual value of a User\_defined\_data\_element (see 4.2.351) or an aggregate thereof.

Each Data\_element\_value is either an Aggregated\_value (see 4.2.7) or a Single\_value (see 4.2.316).

The data associated with a Data\_element\_value are the following:

— name.

##### **4.2.76.1 name**

The name specifies the identifier of the Data\_element\_value.

#### **4.2.77 Date\_and\_person\_assignment**

A Date\_and\_person\_assignment is a relation that associates a Date\_and\_person\_or\_organization (see 4.2.78) with an object.

The data associated with a `Date_and_person_assignment` are the following:

- `assigned_date_and_person`;
- `description`;
- `is_applied_to`;
- `role`.

#### 4.2.77.1 `assigned_date_and_person`

The `assigned_date_and_person` specifies the `Date_and_person_or_organization` (see 4.2.78).

See 4.3.554 for the application assertion.

#### 4.2.77.2 `description`

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Date_and_person_assignment`.

The `description` need not be specified for a particular `Date_and_person_assignment`.

#### 4.2.77.3 `is_applied_to`

The `is_applied_to` specifies the object with which the `Date_and_person_assignment` is associated.

Each `is_applied_to` may be one of the following: `Activity` (see 4.2.1), `Activity_element` (see 4.2.2), `Activity_method_assignment` (see 4.2.4), `Activity_relationship` (see 4.2.5), `Alternate_item_relationship` (see 4.2.11), `Approval_status` (see 4.2.25), `Assembly_component_relationship` (see 4.2.26), `Assembly_substitute_relationship` (see 4.2.28), `Cable_pull_information` (see 4.2.33), `Certification` (see 4.2.38), `Class_category_association` (see 4.2.40), `Class_condition_association` (see 4.2.41), `Class_inclusion_association` (see 4.2.42), `Class_specification_association` (see 4.2.44), `Class_structure_relationship` (see 4.2.45), `Classification_association` (see 4.2.46), `Classification_attribute` (see 4.2.47), `Classification_system` (see 4.2.48), `Complex_product` (see 4.2.51), `Complex_product_relationship` (see 4.2.52), `Composition_relationship` (see 4.2.55), `Configuration` (see 4.2.56), `Connectivity_allocation` (see 4.2.60), `Connectivity_definition` (see 4.2.61), `Connectivity_definition_relationship` (see 4.2.62), `Contract` (see 4.2.63), `Data_element` (see 4.2.70), `Data_element_association` (see 4.2.71), `Data_element_definition` (see 4.2.72), `Data_element_relationship` (see 4.2.74), `Data_element_specification` (see 4.2.75), `Design_discipline_item_definition` (see 4.2.86), `Device` (see 4.2.88), `Device_relationship` (see 4.2.89), `Document` (see 4.2.101), `Document_file` (see 4.2.106), `Document_file_relationship` (see 4.2.107), `Document_representation` (see 4.2.110), `Document_version` (see 4.2.114), `Document_version_relationship` (see 4.2.115), `Drawing` (see 4.2.119), `Drawing_sequence` (see 4.2.121), `Drawing_sheet` (see 4.2.122), `Drawing_sheet_relationship` (see 4.2.124), `Free_segment` (see 4.2.144), `Function_definition` (see 4.2.145), `Function_definition_relationship` (see 4.2.146), `Function_interface` (see 4.2.147), `Function_unit` (see 4.2.148), `Function_unit_relationship` (see 4.2.149), `Function_version` (see 4.2.150), `Function_version_relationship` (see 4.2.151), `Functional_connectivity_definition` (see 4.2.152), `Functional_connectivity_definition_relationship` (see 4.2.153), `Functional_unit_allocation` (see 4.2.154), `Functionality` (see 4.2.155), `General_classification` (see 4.2.156), `Generic_note` (see 4.2.159), `Interface` (see 4.2.170), `Interface_port` (see 4.2.171), `Interface_terminal` (see 4.2.174), `Item` (see 4.2.178), `Item_definition_relationship` (see 4.2.179), `Item_version` (see 4.2.182), `Item_version_relationship` (see 4.2.183), `Location` (see

## ISO 10303-212:2001(E)

4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Organization\_relationship (see 4.2.225), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Person\_in\_organization (see 4.2.238), Person\_in\_organization\_relationship (see 4.2.239), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Port\_association (see 4.2.249), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Requirement\_document\_assignment (see 4.2.287), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.523, 4.3.524, 4.3.525, 4.3.526, 4.3.527, 4.3.528, 4.3.529, 4.3.530, 4.3.531, 4.3.532, 4.3.533, 4.3.534, 4.3.535, 4.3.536, 4.3.537, 4.3.538, 4.3.539, 4.3.540, 4.3.541, 4.3.542, 4.3.543, 4.3.544, 4.3.545, 4.3.546, 4.3.547, 4.3.548, 4.3.549, 4.3.550, 4.3.551, 4.3.552, 4.3.553, 4.3.555, 4.3.556, 4.3.557, 4.3.558, 4.3.559, 4.3.560, 4.3.561, 4.3.562, 4.3.563, 4.3.564, 4.3.565, 4.3.566, 4.3.567, 4.3.568, 4.3.569, 4.3.570, 4.3.571, 4.3.572, 4.3.573, 4.3.574, 4.3.575, 4.3.576, 4.3.577, 4.3.578, 4.3.579, 4.3.580, 4.3.581, 4.3.582, 4.3.583, 4.3.584, 4.3.585, 4.3.586, 4.3.587, 4.3.588, 4.3.589, 4.3.590, 4.3.591, 4.3.592, 4.3.593, 4.3.594, 4.3.595, 4.3.596, 4.3.597, 4.3.598, 4.3.599, 4.3.600, 4.3.601, 4.3.602, 4.3.603, 4.3.604, 4.3.605, 4.3.606, 4.3.607, 4.3.608, 4.3.609, 4.3.610, 4.3.611, 4.3.612, 4.3.613, 4.3.614, 4.3.615, 4.3.616, 4.3.617, 4.3.618, 4.3.619, 4.3.620, 4.3.621, 4.3.622, 4.3.623, 4.3.624, 4.3.625, 4.3.626, 4.3.627, 4.3.628, 4.3.629, 4.3.630, 4.3.631, 4.3.632, 4.3.633, 4.3.634, 4.3.635, 4.3.636, 4.3.637, 4.3.638, 4.3.639, 4.3.640, and 4.3.641 for the application assertions.

### 4.2.77.4 role

The role specifies the relationship between the point in time and the Person (see 4.2.237) or Organization (see 4.2.223) in the Date\_and\_person\_assignment. The value is either user defined or predefined.

The predefined value of role is one of the following:

- creation;
- update.

NOTE See 4.2.77.4.1 - 4.2.77.4.2 for the definition of each predefined value for relation\_type.

#### 4.2.77.4.1 creation

creation: The assignment specifies that the referenced item has been created by the given Person (see 4.2.237) or Organization (see 4.2.223) at the given date and time.

**4.2.77.4.2 update**

update: The assignment specifies that the referenced item has been changed by the given Person (see 4.2.237) or Organization (see 4.2.223) at the given date and time.

**4.2.78 Date\_and\_person\_or\_organization**

A Date\_and\_person\_or\_organization is a Date\_time (see 4.2.79) and a Person (see 4.2.237) or an Organization (see 4.2.223).

The data associated with a Date\_and\_person\_or\_organization are the following:

- associated\_date;
- person\_or\_organization.

**4.2.78.1 associated\_date**

The associated\_date specifies the date component of a Date\_and\_person\_or\_organization.

See 4.3.642 for the application assertion.

**4.2.78.2 person\_or\_organization**

The person\_or\_organization specifies the Organization (see 4.2.223), the Person (see 4.2.237), or the Person\_in\_organization (see 4.2.238) that is part of the Date\_and\_person\_or\_organization.

Each person\_or\_organization may be one of the following: Organization (see 4.2.223) or Person\_in\_organization (see 4.2.238).

See 4.3.643 and 4.3.644 for the application assertions.

**4.2.79 Date\_time**

A Date\_time is the specification of a date and an optional time of day.

The data associated with a Date\_time are the following:

- date;
- time.

**4.2.79.1 date**

The date specifies the calendar time conveying information about year, month, and day.

**4.2.79.2 time**

The time specifies a moment of occurrence measured by hour, minute, and second.

The time need not be specified for a particular Date\_time.

## 4.2.80 Date\_time\_assignment

A Date\_time\_assignment is an association of Date\_time (see 4.2.79) with some product data.

The data associated with a Date\_time\_assignment are the following:

- assigned\_date\_time;
- description;
- is\_applied\_to;
- role.

### 4.2.80.1 assigned\_date\_time

The assigned\_date\_time specifies the Date\_time (see 4.2.79) that is associated with the product data.

See 4.3.676 for the application assertion.

### 4.2.80.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Date\_time\_assignment.

The description need not be specified for a particular Date\_time\_assignment.

### 4.2.80.3 is\_applied\_to

The is\_applied\_to specifies the product data with which the Date\_time\_assignment is associated.

Each is\_applied\_to may be one of the following: Activity (see 4.2.1), Activity\_element (see 4.2.2), Activity\_method\_assignment (see 4.2.4), Activity\_relationship (see 4.2.5), Alternate\_item\_relationship (see 4.2.11), Approval\_status (see 4.2.25), Assembly\_component\_relationship (see 4.2.26), Assembly\_substitute\_relationship (see 4.2.28), Cable\_pull\_information (see 4.2.33), Certification (see 4.2.38), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_association (see 4.2.46), Classification\_attribute (see 4.2.47), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_allocation (see 4.2.60), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Contract (see 4.2.63), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Data\_element\_specification (see 4.2.75), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see 4.2.124), Free\_segment (see 4.2.144), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_



relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), Functionality (see 4.2.155), General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Organization\_relationship (see 4.2.225), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Person\_in\_organization (see 4.2.238), Person\_in\_organization\_relationship (see 4.2.239), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Port\_association (see 4.2.249), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Requirement\_document\_assignment (see 4.2.287), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.645, 4.3.646, 4.3.647, 4.3.648, 4.3.649, 4.3.650, 4.3.651, 4.3.652, 4.3.653, 4.3.654, 4.3.655, 4.3.656, 4.3.657, 4.3.658, 4.3.659, 4.3.660, 4.3.661, 4.3.662, 4.3.663, 4.3.664, 4.3.665, 4.3.666, 4.3.667, 4.3.668, 4.3.669, 4.3.670, 4.3.671, 4.3.672, 4.3.673, 4.3.674, 4.3.675, 4.3.677, 4.3.678, 4.3.679, 4.3.680, 4.3.681, 4.3.682, 4.3.683, 4.3.684, 4.3.685, 4.3.686, 4.3.687, 4.3.688, 4.3.689, 4.3.690, 4.3.691, 4.3.692, 4.3.693, 4.3.694, 4.3.695, 4.3.696, 4.3.697, 4.3.698, 4.3.699, 4.3.700, 4.3.701, 4.3.702, 4.3.703, 4.3.704, 4.3.705, 4.3.706, 4.3.707, 4.3.708, 4.3.709, 4.3.710, 4.3.711, 4.3.712, 4.3.713, 4.3.714, 4.3.715, 4.3.716, 4.3.717, 4.3.718, 4.3.719, 4.3.720, 4.3.721, 4.3.722, 4.3.723, 4.3.724, 4.3.725, 4.3.726, 4.3.727, 4.3.728, 4.3.729, 4.3.730, 4.3.731, 4.3.732, 4.3.733, 4.3.734, 4.3.735, 4.3.736, 4.3.737, 4.3.738, 4.3.739, 4.3.740, 4.3.741, 4.3.742, 4.3.743, 4.3.744, 4.3.745, 4.3.746, 4.3.747, 4.3.748, 4.3.749, 4.3.750, 4.3.751, 4.3.752, 4.3.753, 4.3.754, 4.3.755, 4.3.756, 4.3.757, 4.3.758, 4.3.759, 4.3.760, 4.3.761, 4.3.762, and 4.3.763 for the application assertions.

#### 4.2.80.4 role

The role specifies the action associated with the Date\_time\_assignment. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of role is one of the following:

- classification date;
- creation;
- installation;
- production;
- registration;
- update.

NOTE See 4.2.80.4.1 - 4.2.80.4.6 for the definition of each predefined value for role.

### 4.2.80.4.1 classification date

classification date: The assignment specifies that the specified object is classified at the given date and time. This value shall only be used, if the Date\_time\_assignment refers to instances of Classification\_association (see 4.2.46) as 'is\_applied\_to'.

### 4.2.80.4.2 creation

creation: The assignment specifies that the referenced item was created at the given date and time.

### 4.2.80.4.3 installation

installation: The assignment specifies that the referenced item was mounted in a product at the given date and time.

### 4.2.80.4.4 production

production: The assignment specifies that the referenced item was produced at the given date and time.

### 4.2.80.4.5 registration

registration: The assignment specifies that the referenced item was registered at the given date and time.

### 4.2.80.4.6 update

update: The assignment specifies that the referenced item was changed at the given date and time.

## 4.2.81 Date\_time\_interval\_assignment

A Date\_time\_interval\_assignment is an association of an Interval\_of\_time (see 4.2.177) with some product data.

The data associated with a `Date_time_assignment` (see 4.2.80) are the following:

- `assigned_time_interval`;
- `description`;
- `is_applied_to`;
- `role`.

#### 4.2.81.1 `assigned_time_interval`

The `assigned_time_interval` specifies the `Interval_of_time` (see 4.2.177) that is associated with the product data.

See 4.3.825 for the application assertion.

#### 4.2.81.2 `description`

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Date_time_assignment` (see 4.2.80).

The `description` need not be specified for a particular `Date_time_interval_assignment`.

#### 4.2.81.3 `is_applied_to`

The `is_applied_to` specifies the set of objects of product data with which the `Date_time_interval_assignment` is associated.

There shall be at least one object that the `Date_time_interval_assignment` is assigned to.

Each `is_applied_to` may be one of the following: `Activity` (see 4.2.1), `Activity_element` (see 4.2.2), `Activity_method_assignment` (see 4.2.4), `Activity_relationship` (see 4.2.5), `Alternate_item_relationship` (see 4.2.11), `Approval_status` (see 4.2.25), `Assembly_component_relationship` (see 4.2.26), `Assembly_substitute_relationship` (see 4.2.28), `Cable_pull_information` (see 4.2.33), `Certification` (see 4.2.38), `Class_category_association` (see 4.2.40), `Class_condition_association` (see 4.2.41), `Class_inclusion_association` (see 4.2.42), `Class_specification_association` (see 4.2.44), `Class_structure_relationship` (see 4.2.45), `Classification_association` (see 4.2.46), `Classification_attribute` (see 4.2.47), `Classification_system` (see 4.2.48), `Complex_product` (see 4.2.51), `Complex_product_relationship` (see 4.2.52), `Composition_relationship` (see 4.2.55), `Configuration` (see 4.2.56), `Connectivity_allocation` (see 4.2.60), `Connectivity_definition` (see 4.2.61), `Connectivity_definition_relationship` (see 4.2.62), `Contract` (see 4.2.63), `Data_element` (see 4.2.70), `Data_element_association` (see 4.2.71), `Data_element_definition` (see 4.2.72), `Data_element_relationship` (see 4.2.74), `Data_element_specification` (see 4.2.75), `Design_discipline_item_definition` (see 4.2.86), `Device` (see 4.2.88), `Device_relationship` (see 4.2.89), `Document` (see 4.2.101), `Document_file` (see 4.2.106), `Document_file_relationship` (see 4.2.107), `Document_representation` (see 4.2.110), `Document_version` (see 4.2.114), `Document_version_relationship` (see 4.2.115), `Drawing` (see 4.2.119), `Drawing_sequence` (see 4.2.121), `Drawing_sheet` (see 4.2.122), `Drawing_sheet_relationship` (see 4.2.124), `Free_segment` (see 4.2.144), `Function_definition` (see 4.2.145), `Function_definition_relationship` (see 4.2.146), `Function_interface` (see 4.2.147), `Function_unit` (see 4.2.148), `Function_unit_relationship` (see 4.2.149), `Function_version` (see 4.2.150), `Function_version_relationship` (see 4.2.151), `Functional_connectivity_definition` (see 4.2.152), `Functional_connectivity_definition_`

## ISO 10303-212:2001(E)

relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), Functionality (see 4.2.155), General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Organization\_relationship (see 4.2.225), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Person\_in\_organization (see 4.2.238), Person\_in\_organization\_relationship (see 4.2.239), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Port\_association (see 4.2.249), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Requirement\_document\_assignment (see 4.2.287), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.764, 4.3.765, 4.3.766, 4.3.767, 4.3.768, 4.3.769, 4.3.770, 4.3.771, 4.3.772, 4.3.773, 4.3.774, 4.3.775, 4.3.776, 4.3.777, 4.3.778, 4.3.779, 4.3.780, 4.3.781, 4.3.782, 4.3.783, 4.3.784, 4.3.785, 4.3.786, 4.3.787, 4.3.788, 4.3.789, 4.3.790, 4.3.791, 4.3.792, 4.3.793, 4.3.794, 4.3.795, 4.3.796, 4.3.797, 4.3.798, 4.3.799, 4.3.800, 4.3.801, 4.3.802, 4.3.803, 4.3.804, 4.3.805, 4.3.806, 4.3.807, 4.3.808, 4.3.809, 4.3.810, 4.3.811, 4.3.812, 4.3.813, 4.3.814, 4.3.815, 4.3.816, 4.3.817, 4.3.818, 4.3.819, 4.3.820, 4.3.821, 4.3.822, 4.3.823, 4.3.824, 4.3.826, 4.3.827, 4.3.828, 4.3.829, 4.3.830, 4.3.831, 4.3.832, 4.3.833, 4.3.834, 4.3.835, 4.3.836, 4.3.837, 4.3.838, 4.3.839, 4.3.840, 4.3.841, 4.3.842, 4.3.843, 4.3.844, 4.3.845, 4.3.846, 4.3.847, 4.3.848, 4.3.849, 4.3.850, 4.3.851, 4.3.852, 4.3.853, 4.3.854, 4.3.855, 4.3.856, 4.3.857, 4.3.858, 4.3.859, 4.3.860, 4.3.861, 4.3.862, 4.3.863, 4.3.864, 4.3.865, 4.3.866, 4.3.867, 4.3.868, 4.3.869, 4.3.870, 4.3.871, 4.3.872, 4.3.873, 4.3.874, 4.3.875, 4.3.876, 4.3.877, 4.3.878, 4.3.879, 4.3.880, 4.3.881, and 4.3.882 for the application assertions.

### 4.2.81.4 role

The role specifies the action associated with the Date\_time\_interval\_assignment. The value is either user defined or predefined.

The predefined value of role is one of the following:

— installation.

NOTE See 4.2.81.4.1 for the definition of each predefined value for role.

#### 4.2.81.4.1 installation

installation: The assignment specifies that the referenced object was mounted in a product at the given interval of time.

## 4.2.82 Dated\_configuration

A Dated\_configuration is a type of Manufacturing\_configuration (see 4.2.198) that is a configuration that applies onwards from a given date, or between a start and an end date.

The data associated with a Dated\_configuration are the following:

- end\_date;
- start\_date.

### 4.2.82.1 end\_date

The end\_date specifies the date and time when the validity of the 'configured\_element' is not defined any longer. If the end\_date is not specified, the Dated\_configuration is considered as valid forever from the start\_date.

The end\_date need not be specified for a particular Dated\_configuration.

### 4.2.82.2 start\_date

The start\_date specifies the first date when the Dated\_configuration is valid.

## 4.2.83 Datum\_feature\_callout

A Datum\_feature\_callout is a type of Draughting\_callout (see 4.2.117) that is used to identify a point, line, or plane as a datum and that specifies the designation to be used as identification of that datum.

### 4.2.84 Datum\_target\_callout

A Datum\_target\_callout is a type of Draughting\_callout (see 4.2.117) that is used to identify points, lines, and surfaces of contact, on a part, used in establishing a reference datum. The callout contains an alphanumeric designation and, where applicable, a specification of the diametrical size of the target area.

## 4.2.85 Descriptive\_specification

A Descriptive\_specification is the definitional description of an item.

**NOTE** Usually an item that is specified by a Descriptive\_specification is not represented in the assembly structure of the system.

**EXAMPLE** A Descriptive\_specification can be used to describe the characteristics that distinguish a final part from the corresponding neutral part.

The data associated with a Descriptive\_specification are the following:

## ISO 10303-212:2001(E)

- description;
- id.

### 4.2.85.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Descriptive\_specification.

### 4.2.85.2 id

The id specifies the identifier of the Descriptive\_specification.

The id need not be specified for a particular Descriptive\_specification.

### 4.2.86 Design\_discipline\_item\_definition

A Design\_discipline\_item\_definition is a view of an Item\_version (see 4.2.182) relevant for the requirements of one or more life-cycle stages and application domains. This view collects product data for a specific task.

EXAMPLE Different methods of analysis, such as fault current analysis, voltage drop analysis, or electromagnetic interference analysis, may require distinctive viewpoints. To support this idea, the user may define the subjects of the analyses individually by defining different instances of a Design\_discipline\_item\_definition. Each of these instances refer to the same instance of Item\_version (see 4.2.182).

NOTE 1 The selection of data describing an Item (see 4.2.178) can be different for assembly purposes, packing purposes, or analysis purposes.

NOTE 2 The use of the Design\_discipline\_item\_definition object is not confined to the design stage.

Each Design\_discipline\_item\_definition may be an Assembly\_definition (see 4.2.27).

The data associated with a Design\_discipline\_item\_definition are the following:

- additional\_context;
- associated\_item\_version;
- id;
- initial\_context;
- name.

#### 4.2.86.1 additional\_context

The additional\_context specifies the set of Application\_context (see 4.2.22) objects in which this view of the Item\_version (see 4.2.182) is also relevant. The additional\_context shall not contain the Application\_context (see 4.2.22) that is referenced as the 'initial\_context'.

See 4.3.883 for the application assertion.

#### **4.2.86.2 associated\_item\_version**

The `associated_item_version` specifies the `Item_version` (see 4.2.182) for which the `Design_discipline_item_definition` is a view.

See 4.3.885 for the application assertion.

#### **4.2.86.3 id**

The `id` specifies the identifier of the `Design_discipline_item_definition`.

#### **4.2.86.4 initial\_context**

The `initial_context` specifies the `Application_context` (see 4.2.22) in which this view of the `Item_version` (see 4.2.182) has been designed primarily.

See 4.3.884 for the application assertion.

#### **4.2.86.5 name**

The `name` specifies a speaking designation of the `Design_discipline_item_definition`.

The `name` need not be specified for a particular `Design_discipline_item_definition`.

### **4.2.87 Detached\_representation\_reference**

A `Detached_representation_reference` is a type of `Cross_reference` (see 4.2.64) that is a reference made from one part of a diagram to another part between the different presentations of the electrotechnical system. Using references is a widely used technique to keep diagrams clear.

EXAMPLE Reference among the detached representations and the attached representation of a piece of equipment.

The data associated with a `Detached_representation_reference` are the following:

- `part_of`;
- `refers_to`.

#### **4.2.87.1 part\_of**

The `part_of` specifies the `Annotation_element` (see 4.2.15) the `Detached_representation_reference` belongs to.

See 4.3.886 for the application assertion.

## 4.2.87.2 refers\_to

The refers\_to specifies the Annotation\_element (see 4.2.15) the Detached\_representation\_reference references.

See 4.3.887 for the application assertion.

## 4.2.88 Device

A Device is a type of Product\_constituent (see 4.2.266) that is an occurrence of a Design\_discipline\_item\_definition (see 4.2.86) object.

NOTE 1 A Device may be instantiated more than once since each instance is an individual occurrence of the piece of equipment that is characterized by the Design\_discipline\_item\_definition (see 4.2.86).

EXAMPLE 1 In a specific circuit, the Item (see 4.2.178) 'lamp' is defined once. This Design\_discipline\_item\_definition (see 4.2.86) carries all the information defining the lamp (e.g., its terminals) that is independent from its usage. Additionally, three Device objects for this Design\_discipline\_item\_definition (see 4.2.86) exist because three equal lamps are used within this particular circuit. Each of these instances can be connected individually.

Each Device is either a Quantified\_device (see 4.2.276), a Selected\_device (see 4.2.303), a Single\_device (see 4.2.314), or a Specified\_device (see 4.2.328).

The data associated with a Device are the following:

- definition;
- description;
- extended\_designation;
- id.

### 4.2.88.1 definition

The definition specifies the reference to the associated defining Design\_discipline\_item\_definition (see 4.2.86) or Product\_identification (see 4.2.268) object.

Each definition may be one of the following: Design\_discipline\_item\_definition (see 4.2.86) or Product\_identification (see 4.2.268).

See 4.3.888 and 4.3.890 for the application assertions.

### 4.2.88.2 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Device.

The description need not be specified for a particular Device.



### 4.2.88.3 extended\_designation

The extended\_designation specifies a structured label for the Device.

NOTE 2 The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

EXAMPLE 2 IEC 61346-1 specifies designations and structuring principles for devices.

The extended\_designation need not be specified for a particular Device.

See 4.3.889 for the application assertion.

### 4.2.88.4 id

The id specifies an identifier for the Device.

## 4.2.89 Device\_relationship

A Device\_relationship is the relation between two Device (see 4.2.88) objects.

The data associated with an Device\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.89.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Device\_relationship.

The description need not be specified for a particular Device\_relationship.

### 4.2.89.2 related

The related specifies the second of the two Device (see 4.2.88) objects related by the Device\_relationship.

See 4.3.891 for the application assertion.

### 4.2.89.3 relating

The relating specifies the first of the two Device (see 4.2.88) objects related by the Device\_relationship.

See 4.3.892 for the application assertion.

#### 4.2.89.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- associated equipment;
- derivation;
- envelope;
- redundancy;
- shield;
- substitution.

NOTE See 4.2.89.4.1 - 4.2.89.4.6 for the definition of each predefined value for relation\_type.

##### 4.2.89.4.1 associated equipment

associated equipment: The Device\_relationship defines a relationship where the related Device (see 4.2.88) is equipment that is assigned to the related Device (see 4.2.88) with the intention to support it in performing its task.

EXAMPLE 1 The association of the strain relief to a connector during the installation process.

##### 4.2.89.4.2 derivation

derivation: The Device\_relationship defines a deriving relationship where the related Device (see 4.2.88) is based on the relating Device (see 4.2.88).

##### 4.2.89.4.3 envelope

envelope: The Device\_relationship defines a relationship where the related Device (see 4.2.88) is enclosed by the relating Device (see 4.2.88).

EXAMPLE 2 Using an overbraid or taping are methods to envelop wires.

##### 4.2.89.4.4 redundancy

redundancy: The Device\_relationship defines a relationship where the related Devices replicated by the relating Device (see 4.2.88).

EXAMPLE 3 To provide for a fail-safe service two drives do the work that could be done by one motor. If one drive fails, the other is still in service.

##### 4.2.89.4.5 shield

shield: The Device\_relationship defines a relationship where the relating Device (see 4.2.88) provides the screen against electromagnetic interference. The related Device (see 4.2.88) is considered to be

inside the screen. If more than one shield is present, there shall be a Device\_relationship where the relating Device (see 4.2.88) is the outer shield and the related Device (see 4.2.88) is the inner shield.

NOTE In cases when the screens do not enclose the shielded hardware the 'description' attribute shall specify the arrangement of the screens.

#### 4.2.89.4.6 substitution

substitution: The Device\_relationship defines a relationship where the related Device (see 4.2.88) replaces the relating Device (see 4.2.88).

### 4.2.90 Diameter\_dimension

A Diameter\_dimension is a type of Dimension (see 4.2.93) that is the graphical presentation of the value of the diametrical size of a circular element.

The data associated with a Diameter\_dimension are the following:

- component;
- extent.

#### 4.2.90.1 component

The component specifies the projection lines that shows the extension of the points, lines, or surfaces that bound the measurement.

See 4.3.894 for the application assertion.

#### 4.2.90.2 extent

The extent specifies the dimension line that graphically presents where the dimension value applies.

See 4.3.893 for the application assertion.

### 4.2.91 Digital\_document

A Digital\_document is a type of Document\_representation (see 4.2.110) that is a set of data in electronic form that is intended to give further information about the product data.

NOTE The content of the Digital\_document need not be specified by using any part of ISO 10303.

EXAMPLE A Digital\_document contains native data from applications such as desktop publishing systems or simulation systems.

The data associated with a Digital\_document are the following:

- file.

#### 4.2.91.1 file

## ISO 10303-212:2001(E)

The file specifies the `Digital_file` (see 4.2.92) that contains the data.

See 4.3.895 for the application assertion.

### 4.2.92 `Digital_file`

The `Digital_file` is a type of `Document_file` (see 4.2.106) that is a collection of related data stored under one name.

### 4.2.93 `Dimension`

A `Dimension` is a type of `Draughting_annotation` (see 4.2.116) that is the graphical presentation of a dimension value, associated information, and the necessary symbology to accurately depict its area of application.

Each `Dimension` is either an `Angular_dimension` (see 4.2.13), a `Curve_dimension` (see 4.2.69), a `Diameter_dimension` (see 4.2.90), a `Leader_directed_dimension` (see 4.2.187), a `Linear_dimension` (see 4.2.190), an `Ordinate_dimension` (see 4.2.222), or a `Radius_dimension` (see 4.2.277).

### 4.2.94 `Dimension_callout`

A `Dimension_callout` is the text and symbols in the presentation of a dimension that represent the dimension value, dimension units, tolerance information, and any related notes.

Each `Dimension_callout` is either a `Structured_dimension_callout` (see 4.2.332) or a `Unstructured_dimension_callout` (see 4.2.349).

The data associated with a `Dimension_callout` are the following:

- `defined_primary_dimension`;
- `defined_secondary_dimension`.

#### 4.2.94.1 `defined_primary_dimension`

The `defined_primary_dimension` specifies the dimension in which the dimension value, tolerance information, and any associated notes of the `Dimension_callout` are given in the primary unit of measure for the drawing view, drawing sheet, or drawing in which the dimension appears.

The `defined_primary_dimension` need not be specified for a particular `Dimension_callout`.

See 4.3.896 for the application assertion.

#### 4.2.94.2 `defined_secondary_dimension`

The `defined_secondary_dimension` specifies the dimension in which the dimension value, tolerance information, and any associated notes of the `Dimension_callout` are given in a different unit of measure than that given for the drawing view, drawing sheet, or drawing in which the dimension appears.

The `defined_secondary_dimension` need not be specified for a particular `Dimension_callout`.

See 4.3.897 for the application assertion.

### 4.2.95 Dimension\_line

A Dimension\_line is a type of Directed\_curve (see 4.2.99) that is used in the graphical presentation of a dimension value along with other symbology, if necessary, to show the extent of the application of the value.

EXAMPLE Terminator symbols may be associated with a Dimension\_line.

### 4.2.96 Dimension\_line\_terminator

A Dimension\_line\_terminator is an annotation symbol that is applied to a dimension line and used to identify the endpoint or point of the application of the directed annotation.

The data associated with a Dimension\_line\_terminator are the following:

- line;
- symbol.

#### 4.2.96.1 line

The line specifies the Dimension\_line (see 4.2.95) that the symbol applies to.

See 4.3.899 for the application assertion.

#### 4.2.96.2 symbol

The symbol specifies an Annotation\_symbol (see 4.2.20) that is used to identify the endpoint or point of application of the directed annotation.

See 4.3.898 for the application assertion.

### 4.2.97 Dimension\_sequence\_pair

A Dimension\_sequence\_pair is the relationship between two adjacent dimensions that share a projection line.

Each Dimension\_sequence\_pair is either a Chained\_dimension\_pair (see 4.2.39) or a Parallel\_dimension\_pair (see 4.2.230).

The data associated with a Dimension\_sequence\_pair are the following:

- predecessor;
- successor.

#### 4.2.97.1 predecessor

The predecessor specifies the dimension that is displayed first in a dimension sequence.

## ISO 10303-212:2001(E)

See 4.3.900 for the application assertion.

### 4.2.97.2 successor

The successor specifies the dimension that is displayed second in a dimension sequence.

See 4.3.901 for the application assertion.

### 4.2.98 Dimension\_symbol

A Dimension\_symbol is a type of Predefined\_symbol (see 4.2.255) that is used in conjunction with a dimension value to convey the context of the dimension value.

EXAMPLE A diameter symbol may be used in conjunction with a dimension value to denote a diameter dimension.

The predefined dimension symbols that shall be supported by all implementations of this part of ISO 10303 are dependent on the height (h) of the text that the symbol accompanies. The height of the characters for the predefined symbols specified here shall be  $h = 2.5$  mm. The illustrations shown in Figure 12 are oriented as they appear when associated with a horizontal dimension line.

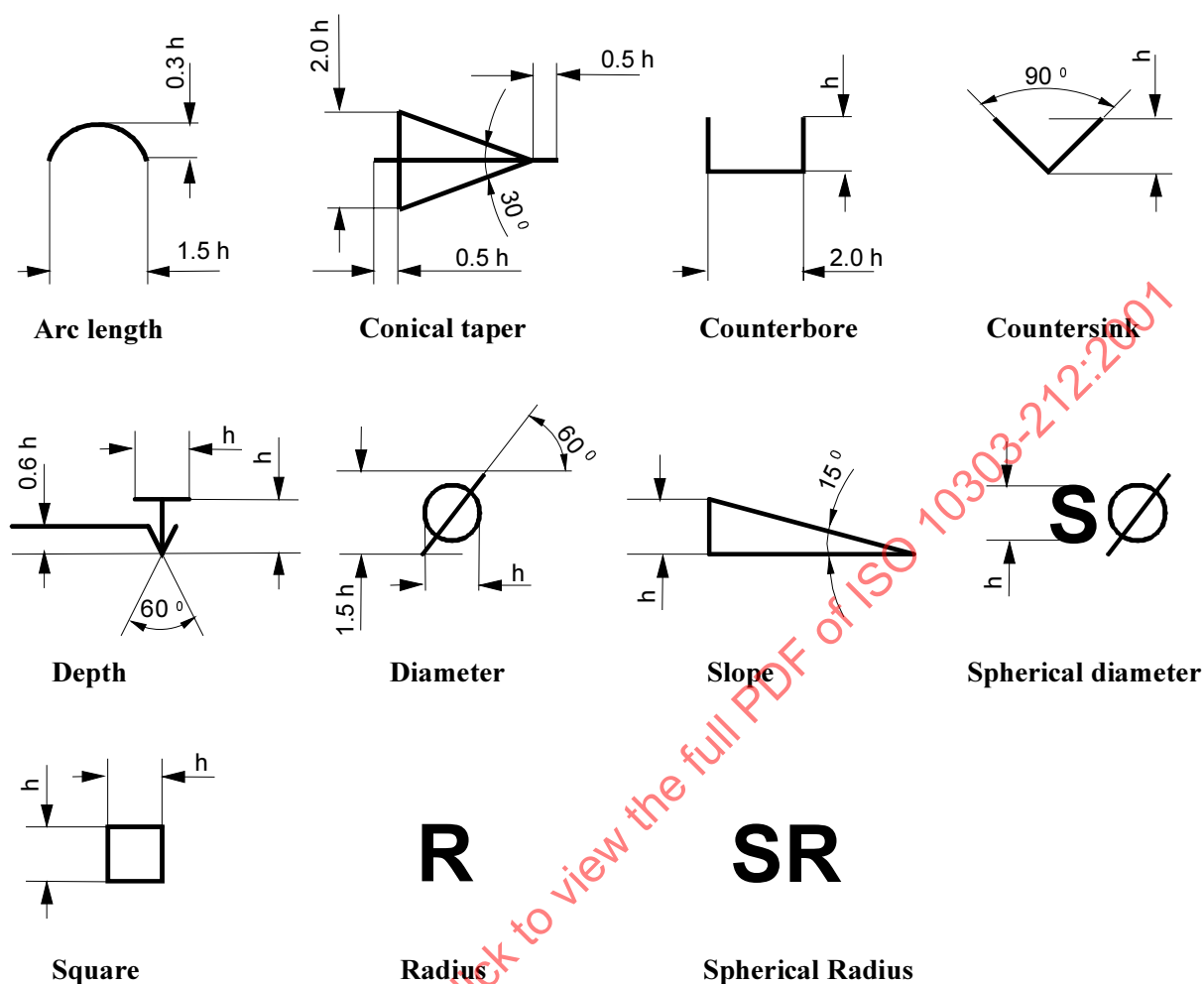


Figure 12 - Predefined dimension symbols

The data associated with a Dimension\_symbol are the following:

— symbol\_type.

#### 4.2.98.1 symbol\_type

The symbol\_type specifies an alphanumeric string identifying the Dimension\_symbol in accordance with the definitions given above.

## ISO 10303-212:2001(E)

The value of symbol\_type is one of the following:

- arc length;
- conical taper;
- counterbore;
- countersink;
- depth;
- diameter;
- slope;
- spherical diameter;
- square;
- radius;
- spherical radius.

NOTE See 4.2.98.1.1 - 4.2.98.1.11 for the definition of each permissible value for symbol\_type.

### 4.2.98.1.1 arc length

arc length: An arc length symbol is a graphical symbol used in conjunction with a dimension value to indicate that the curve dimension is an arc length measured along the curved line or surface. An arc length symbol is depicted as one line segment that forms a part of a circle. The origin of the symbol corresponds to the starting point on the left of the arc line. The size and graphical representation of the symbol are shown in Figure 12.

### 4.2.98.1.2 conical taper

conical taper: A conical taper symbol is a graphical symbol used in conjunction with a dimension value to indicate that the value given is the ratio of the difference in the diameters of two sections taken perpendicular to the datum axis of a cone and the distance between the two sections. A conical taper symbol is depicted as an isosceles triangle and a line placed as a bisector to the angle created by the two equal sides. The origin of the symbol corresponds to the intersection point of the two equal sides. The size and graphical representation of the symbol are shown in Figure 12.

### 4.2.98.1.3 counterbore

counterbore: A counterbore symbol is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value specifies the size of a counterbored hole. A counterbored hole is a cylindrical hole of larger diameter than its basis hole. Both holes share a common centre axis and the bottom of the counterbored hole is planar and perpendicular to the centre axis. A counterbore symbol is depicted as an open rectangle. The origin of the symbol is the geometrical centre of the rectangle. The size and graphical representation of the symbol are shown in Figure 12.



**4.2.98.1.4 countersink**

countersink: A countersink symbol is a graphical symbol used in conjunction with a dimension value to indicate that the associated dimension value applies to a countersink that is concentric to a basis hole. A countersink is a conical taper detail located at the end of an existing or basis hole. The diameter of the taper detail is larger than the diameter of the basis hole at the surface and decreases at a constant rate as a function of distance along the centre axis of the basis hole until the two diameters are equal. A countersink is depicted as an open triangle. The origin of the symbol is the intersection point of the two visible sides. The size and graphical representation of the symbol are shown in Figure 12.

**4.2.98.1.5 depth**

depth: A depth symbol is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value applies to the depth of the feature. A depth symbol is depicted as an arrow with a line placed perpendicular to the end opposite the arrowhead. The origin of the symbol is the top of the arrowhead. The size and graphical representation of the symbol are shown in Figure 12.

**4.2.98.1.6 diameter**

diameter: A diameter symbol is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value applies diametrically. A diameter is depicted as a circle crossed by a line going through the centre. The origin of the symbol is the centre point of the circle. The size and graphical representation of the symbol are shown in Figure 12.

**4.2.98.1.7 slope**

slope: A slope symbol is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value specifies the ratio of the change in the vertical direction to the change in horizontal direction. A slope symbol is depicted as a right triangle. The origin of the symbol is the intersection of the two perpendicular sides of the triangle. The size and graphical representation of the symbol are shown in Figure 12.

**4.2.98.1.8 spherical diameter**

spherical diameter: A spherical diameter symbol is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value applies to the surface or surfaces. A spherical diameter is depicted as a diameter symbol as noted above and the letter 'S' before it. The origin of the symbol is the geometrical centre of an imaginary surrounding box around the whole symbol. The size and graphical representation of the symbol are shown in Figure 12.

**4.2.98.1.9 square**

square: A square symbol is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value applies linearly to two orthogonal geometric elements that form two adjacent sides of a square feature. A square is depicted as a rectangle. The origin of the symbol is the geometrical centre of the rectangle. The size and graphical representation of the symbol are shown in Figure 12.

#### 4.2.98.1.10 radius

radius: A radius symbol is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value applies radially. A radius is depicted by the letter 'R'. The origin of the symbol is the origin of the letter. The size and graphical representation of the symbol are shown in Figure 12.

#### 4.2.98.1.11 spherical radius

spherical radius: A spherical radius is a graphical symbol used in conjunction with a dimension value to indicate that the dimension value applies radially to all points on the dimensioned surface. A spherical radius is depicted by the letters 'S' and 'R'. The origin of the symbol is the geometrical centre of an imaginary box surrounding the whole symbol. The size and graphical representation of the symbol are shown in Figure 12.

### 4.2.99 Directed\_curve

A Directed\_curve is a type of Annotation\_curve (see 4.2.14) that is used to guide annotation to a specific feature or area of a drawing view or drawing sheet.

Each Directed\_curve is either a Dimension\_line (see 4.2.95), a Leader (see 4.2.186), or a Projection\_line (see 4.2.273).

The data associated with a Directed\_curve are the following:

— directed\_callout.

#### 4.2.99.1 directed\_callout

The directed\_callout specifies the Draughting\_callout (see 4.2.117) that is directed by means of the curve.

The directed\_callout need not be specified for a particular Directed\_curve.

See 4.3.902 for the application assertion.

### 4.2.100 Direction\_range

A Direction\_range is the specification of the sector under which connecting lines are allowed to be drawn onto a Schematic\_node (see 4.2.294).

NOTE The information provided by the `Direction_range` can be used by a routing algorithm to place connection lines automatically on a schematic diagram.

The data associated with a `Direction_range` are the following:

- `associated_connect_area`;
- `maximum_angle`;
- `minimum_angle`.

#### **4.2.100.1 associated\_connect\_area**

The `associated_connect_area` specifies the `Connect_area` (see 4.2.57) object that may be approached under the given angles.

See 4.3.903 for the application assertion.

#### **4.2.100.2 maximal\_angle**

The `maximal_angle` specifies the end angle of the sector. The angles are defined in relation to the coordinate space of the symbol.

#### **4.2.100.3 minimal\_angle**

The `minimal_angle` specifies the start angle of the sector. The angles are defined in relation to the coordinate space of the symbol.

### **4.2.101 Document**

A Document is the reference to digital data or nondigital data that are not within the scope of ISO 10303.

The data associated with a Document are the following:

- `description`;
- `extended_designation`;
- `id`;
- `name`.

#### **4.2.101.1 description**

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the Document.

The `description` need not be specified for a particular Document.

#### **4.2.101.2 extended\_designation**

## ISO 10303-212:2001(E)

The `extended_designation` specifies a structured label for the Document.

NOTE The label assigned through `extended_designation` shall be identical to the label assigned by the 'id' attribute.

EXAMPLE IEC 62023 specifies designations and structuring principles for documents.

The `extended_designation` need not be specified for a particular Document.

See 4.3.904 for the application assertion.

### 4.2.101.3 id

The `id` specifies the identifier for the Document.

### 4.2.101.4 name

The `name` specifies a speaking designation of the Document.

The `name` need not be specified for a particular Document.

## 4.2.102 Document\_assignment

A `Document_assignment` is a relation that associates a `Document_version` (see 4.2.114) with an item.

The data associated with a `Document_assignment` are the following:

- `assigned_document`;
- `is_assigned_to`;
- `role`.

### 4.2.102.1 assigned\_document

The `assigned_document` specifies the `Document_version` (see 4.2.114).

Each `assigned_document` may be one of the following: `Document` (see 4.2.101), `Document_file` (see 4.2.106), `Document_representation` (see 4.2.110), or `Document_version` (see 4.2.114).

See 4.3.930, 4.3.931, 4.3.932, and 4.3.933 for the application assertions.

### 4.2.102.2 is\_assigned\_to

The `is_assigned_to` specifies the item with which the `Document_version` (see 4.2.114) is associated.

Each `is_assigned_to` may be one of the following: `Activity` (see 4.2.1), `Activity_element` (see 4.2.2), `Activity_method` (see 4.2.3), `Address` (see 4.2.6), `Approval` (see 4.2.23), `Approval_status` (see 4.2.25), `Assembly_component_relationship` (see 4.2.26), `Cable_pull_information` (see 4.2.33), `Certification` (see 4.2.38), `Class_category_association` (see 4.2.40), `Class_condition_association` (see 4.2.41), `Class_inclusion_association` (see 4.2.42), `Class_specification_association` (see 4.2.44), `Classification_association` (see 4.2.46), `Classification_attribute` (see 4.2.47), `Classification_system` (see 4.2.48), `Complex_product` (see 4.2.51), `Connectivity_definition` (see 4.2.61), `Contract` (see

4.2.63), Data\_element (see 4.2.70), Data\_element\_definition (see 4.2.72), Data\_element\_specification (see 4.2.75), Descriptive\_specification (see 4.2.85), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Drawing (see 4.2.119), Drawing\_sheet (see 4.2.122), Drawing\_view (see 4.2.125), Function\_definition (see 4.2.145), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_version (see 4.2.150), Functional\_connectivity\_definition (see 4.2.152), Functionality (see 4.2.155), General\_classification (see 4.2.156), General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_identification (see 4.2.180), Item\_version (see 4.2.182), Location (see 4.2.192), Marking (see 4.2.199), Node (see 4.2.208), Notification (see 4.2.213), Object\_designation (see 4.2.217), Organization (see 4.2.223), Path (see 4.2.232), Path\_node (see 4.2.233), Person (see 4.2.237), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Process\_variable (see 4.2.260), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Retention\_period (see 4.2.289), Route (see 4.2.290), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.905, 4.3.906, 4.3.907, 4.3.908, 4.3.909, 4.3.910, 4.3.911, 4.3.912, 4.3.913, 4.3.914, 4.3.915, 4.3.916, 4.3.917, 4.3.918, 4.3.919, 4.3.920, 4.3.921, 4.3.922, 4.3.923, 4.3.924, 4.3.925, 4.3.926, 4.3.927, 4.3.928, 4.3.929, 4.3.934, 4.3.935, 4.3.936, 4.3.937, 4.3.938, 4.3.939, 4.3.940, 4.3.941, 4.3.942, 4.3.943, 4.3.944, 4.3.945, 4.3.946, 4.3.947, 4.3.948, 4.3.949, 4.3.950, 4.3.951, 4.3.952, 4.3.953, 4.3.954, 4.3.955, 4.3.956, 4.3.957, 4.3.958, 4.3.959, 4.3.960, 4.3.961, 4.3.962, 4.3.963, 4.3.964, 4.3.965, 4.3.966, 4.3.967, 4.3.968, 4.3.969, 4.3.970, 4.3.971, 4.3.972, 4.3.973, 4.3.974, 4.3.975, 4.3.976, 4.3.977, 4.3.978, 4.3.979, 4.3.980, 4.3.981, 4.3.982, 4.3.983, 4.3.984, 4.3.985, and 4.3.986 for the application assertions.

### 4.2.102.3 role

The role describes the relationship between the Document\_version (see 4.2.114) and the associated item. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of role is one of the following:

- additional information;
- behaviour;
- catalogue;
- description;
- informative;
- mandatory;
- manual;
- mathematical description;
- specification.

NOTE See 4.2.102.3.1 - 4.2.102.3.9 for the definition of each predefined value for role.

### 4.2.102.3.1 additional information

additional information: The assigned document provides explanatory information.

### 4.2.102.3.2 behaviour

behaviour: The assigned document specifies information about the behaviour of the object specified by the 'is\_assigned\_to' attribute.

EXAMPLE State transition charts or flow diagrams can be used to provide information about the behaviour of an item.

### 4.2.102.3.3 catalogue

catalogue: The assigned document is the catalogue in which the object specified by the 'is\_assigned\_to' attribute is listed.

EXAMPLE The document can be the catalogue of the manufacturer.

### 4.2.102.3.4 description

description: The assigned document provides textual information about the object specified by the 'is\_assigned\_to' attribute.

### 4.2.102.3.5 informative

informative: The assigned document is for information only. It may or may not be considered.

### 4.2.102.3.6 mandatory

manual: The assigned element shall conform to the content of the assigned document.

**4.2.102.3.7 mathematical description**

mathematical description: The assigned document specifies the associated item by providing the algorithmic specification of its behaviour.

EXAMPLE The information can be used for simulation purposes.

**4.2.102.3.8 specification**

specification: The assigned document the considerations that led to the actual design of the object specified by the 'is\_assigned\_to' attribute.

**4.2.103 Document\_content\_property**

A Document\_content\_property specifies characteristics precisising the content of a Document\_file (see 4.2.106) or of a Document\_representation (see 4.2.110). At least one of the optional attributes shall be specified for each instance of this object.

In the case where a Document\_content\_property is referred by a Document\_representation (see 4.2.110), the characteristics apply to all individual Document\_file (see 4.2.106) objects, whereas in the case where it is referred by a Document\_file (see 4.2.106), the characteristics apply on an individual basis.

The data associated with a Document\_content\_property are the following:

- detail\_level;
- geometry\_type;
- languages;
- real\_world\_scale.

**4.2.103.1 detail\_level**

The detail\_level specifies the level of detail that the Document\_file (see 4.2.106) or the Document\_representation (see 4.2.110) provides. The value is either user defined or predefined.

The predefined value of detail\_level is one of the following:

- rough 3d shape;
- rounded edges.

NOTE See 4.2.103.1.1 - 4.2.103.1.2 for the definition of each predefined value for detail\_level.

**4.2.103.1.1 rough 3d shape**

rough 3d shape: 3D shape model without edge rounds and fillets.

**4.2.103.1.2 rounded edges**

rounded edges: 3D shape model with edge rounds and fillets.

## ISO 10303-212:2001(E)

The detail\_level need not be specified for a particular Document\_content\_property.

### 4.2.103.2 geometry\_type

The geometry\_type specifies the kind or kinds of geometry that an object contains.

The value is either user defined or predefined.

The predefined value of detail\_level is one of the following:

- 3D wireframe model;
- 2D shape;
- surface model;
- closed volume;
- solid model;
- solid and surface model;
- assembly;
- assembly with mating elements;
- 2D drawing;
- drawing derived from 3D data;
- drawing related to 3D data.

NOTE See 4.2.103.2.1 - 4.2.103.2.11 for the definition of each predefined value for detail\_level.

#### 4.2.103.2.1 3D wireframe model

3D wireframe model: The document contains a 3D shape model in wireframe representation.

#### 4.2.103.2.2 2D shape

2D shape: The document contains a 2D shape model or contours only.

#### 4.2.103.2.3 surface model

surface model: The document contains a 3D shape model in surface representation.

#### 4.2.103.2.4 closed volume

closed volume: The document contains a 3D shape model in closed body topological surface representation.

#### 4.2.103.2.5 solid model



solid model: The document contains a 3D shape model in advanced boundary representation.

#### **4.2.103.2.6 solid and surface model**

solid and surface model: The document contains a 3D shape model in surface and advanced boundary representation.

#### **4.2.103.2.7 assembly**

assembly: The document contains an assembly structure with reference to the assembled components and their transformation matrices.

#### **4.2.103.2.8 assembly with mating elements**

assembly with mating elements: The document contains an assembly structure including the mating components only, such as screws or rivets, with exact positioning information. This assembly representation is intended to be overlaid with the assembly structure for the main components.

#### **4.2.103.2.9 2D drawing**

2D drawing: The document contains a technical drawing without 3D shape representation.

#### **4.2.103.2.10 drawing derived from 3D data**

drawing derived from 3D data: : The document contains a technical drawing that has been derived from a 3D shape model.

#### **4.2.103.2.11 drawing related to 3D data**

drawing related to 3D data: : The document contains a technical drawing that visualizes a 3D shape model and possibly establishes associative links to the 3D shape model.

The geometry\_type need not be specified for a particular Document\_content\_property.

### **4.2.103.3 languages**

The languages specifies which language or languages are used in the characterized objects.

EXAMPLE 'Japanese' and 'German' are examples for the fact that annotation on a drawing is provided in the language 'Japanese' or 'German' respectively.

See 4.3.987 for the application assertion.

### **4.2.103.4 real\_world\_scale**

The real\_world\_scale specifies the scale that is used in the Document\_file (see 4.2.106) or in the Document\_representation (see 4.2.110) the Document\_content\_property is referred by.

The real\_world\_scale need not be specified for a particular Document\_content\_property.

See 4.3.988 for the application assertion.

## 4.2.104 Document\_creation\_property

A Document\_creation\_property specifies characteristics of Document\_file (see 4.2.106) or of Document\_representation (see 4.2.110). It specifies the context of the creation of the object. At least one of the optional attributes shall be specified for each instance of this object.

In the case where a Document\_creation\_property is referred by a Document\_representation (see 4.2.110) the characteristics apply to all individual Document\_file (see 4.2.106) objects, whereas in the case it is referred by a Document\_file (see 4.2.106), the characteristics apply on an individual basis.

The data associated with a Document\_creation\_property are the following:

- creating\_interface;
- creating\_system;
- operating\_system.

### 4.2.104.1 creating\_interface

The creating\_interface specifies the computer application used to create the Document\_file (see 4.2.106) or Document\_representation (see 4.2.110) object.

EXAMPLE 1 'CATIGE V4.1.8' is an example for a creating interface of a Digital\_document (see 4.2.91).

EXAMPLE 2 'Postscript Printer Driver' is an example for a creating interface of a Physical\_document (see 4.2.242).

EXAMPLE 3 'SYSTEM-C-STL' is an example for a creating interface of a Physical\_model (see 4.2.244) in the case of a stereolithographic model.

The creating\_interface need not be specified for a particular Document\_creation\_property.

### 4.2.104.2 creating\_system

The creating\_system specifies the computer application or the machine which is used to create the object that is characterized.

### 4.2.104.3 operating\_system

The operating\_system specifies the operating system that is used to execute the computer application that created the characterized object.

The operating\_system need not be specified for a particular Document\_creation\_property.

## 4.2.105 Document\_designation

A Document\_designation is a type of Object\_designation (see 4.2.217) that is an identifier of a document.

EXAMPLE The title information of a document can be used as Document\_designation.

### 4.2.106 Document\_file

A Document\_file is one of potentially more files on a computer system or in actual stacks of paper that make up a Document\_representation (see 4.2.110).

Each Document\_file is either a Digital\_file (see 4.2.92) or a Hardcopy (see 4.2.166).

The data associated with a Document\_file are the following:

- content;
- creation;
- description;
- document\_file\_type;
- external\_id\_and\_location;
- file\_format;
- id;
- size;
- version\_id.

#### 4.2.106.1 content

The content characterizes the content of the Document\_file.

The content need not be specified for a particular Document\_file.

See 4.3.989 for the application assertion.

#### 4.2.106.2 creation

The creation specifies further details of the context of the creation of the Document\_file.

The creation need not be specified for a particular Document\_file.

See 4.3.990 for the application assertion.

#### 4.2.106.3 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Document\_file.

The description need not be specified for a particular Document\_file.

#### **4.2.106.4 document\_file\_type**

The document\_file\_type specifies the format of the Document\_file. It shall only be specified, if the Document\_file does not participate in a Document (see 4.2.101).

The document\_file\_type need not be specified for a particular Document\_file.

See 4.3.993 for the application assertion.

#### **4.2.106.5 external\_id\_and\_location**

The external\_id\_and\_location specifies alternatives of the identifier and location of the Document\_file.

EXAMPLE A copy of a document may be found in another department with an different id.

See 4.3.994 for the application assertion.

#### **4.2.106.6 file\_format**

The file\_format specifies the characteristics of the Document\_file that specify the format of the object.

The file\_format need not be specified for a particular Document\_file.

See 4.3.991 for the application assertion.

#### **4.2.106.7 id**

The id specifies the identifier which is used to locate the file either on a computer system or in a repository of paper documents.

#### **4.2.106.8 size**

The size specifies characteristics for the size of the Document\_file.

The size need not be specified for a particular Document\_file.

See 4.3.992 for the application assertion.

#### **4.2.106.9 version\_id**

The version\_id specifies the identification of the version that distinguishes one Document\_file object from other versions of Document\_file objects with the same id.

The version\_id need not be specified for a particular Document\_file.

### 4.2.107 Document\_file\_relationship

A Document\_file\_relationship is a relationship between two Document\_file (see 4.2.106) objects. It specifies that the related Document\_file (see 4.2.106) is referenced from the relating Document\_file (see 4.2.106).

The data associated with a Document\_file\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### 4.2.107.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Document\_file\_relationship.

The description need not be specified for a particular Document\_file\_relationship.

#### 4.2.107.2 related

The related specifies the second of the two Document\_file (see 4.2.106) objects related by the Document\_file\_relationship.

See 4.3.995 for the application assertion.

#### 4.2.107.3 relating

The relating specifies the first of the two Document\_file (see 4.2.106) objects related by the Document\_file\_relationship.

See 4.3.996 for the application assertion.

#### 4.2.107.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of relation\_type is one of the following:

- addition;
- copy;
- decomposition;
- derivation;
- peer;
- reference;
- sequence;
- substitution;
- translation.

NOTE See 4.2.107.4.1 - 4.2.107.4.8 for the definition of each predefined value for relation\_type.

### 4.2.107.4.1 addition

addition: The Document\_file\_relationship specifies that the related document provides supplementary or collateral information with regard to the information provided by the relating document.

### 4.2.107.4.2 copy

copy: The Document\_file\_relationship defines a relationship where the related Document\_representation (see 4.2.110) is a copy of the relating Document\_representation (see 4.2.110).

### 4.2.107.4.3 decomposition

decomposition: The Document\_file\_relationship defines a relationship where the related Document\_representation (see 4.2.110) is one of potentially more sub documents of the relating Document\_representation (see 4.2.110).

### 4.2.107.4.4 derivation

derivation: The Document\_file\_relationship defines a relationship where the related Document\_representation (see 4.2.110) is derived from the relating Document\_representation (see 4.2.110).

### 4.2.107.4.5 reference

reference: The application object defines a relationship where the related document is referenced from the relating.

#### 4.2.107.4.6 sequence

sequence: The Document\_file\_relationship defines a logical sequence where the related Document\_representation (see 4.2.110) comes after the relating Document\_representation (see 4.2.110) (e.g. a sequence of clauses).

#### 4.2.107.4.7 substitution

substitution: The Document\_file\_relationship defines a relationship where the related Document\_representation (see 4.2.110) replaces the relating Document\_representation (see 4.2.110).

#### 4.2.107.4.8 translation

translation: The Document\_file\_relationship specifies that the related document is generated through a translation process from the relating document.

### 4.2.108 Document\_format\_property

A Document\_format\_property specifies characteristics of a Document\_file (see 4.2.106) or of a Document\_representation (see 4.2.110) that specify the format of the object. At least one of the optional attributes shall be specified for each instance of this object.

In the case where a Document\_format\_property is referred by a Document\_representation (see 4.2.110), the characteristics apply to all individual Document\_file (see 4.2.106) objects, whereas in the case it is referred by a Document\_file (see 4.2.106) the characteristics apply on an individual basis.

The data associated with a Document\_representation (see 4.2.110) are the following:

- character\_code;
- data\_format;
- size\_format.

#### 4.2.108.1 character\_code

The character\_code specifies the character code that is used in the characterized object.

## ISO 10303-212:2001(E)

The predefined value of character\_code is one of the following:

- binary;
- IEC 61286;
- ISO 646;
- ISO 3098-1;
- ISO 6937;
- ISO 8859-1;
- ISO 10646.

NOTE 1 See 4.2.108.1.1 - 4.2.108.1.7 for the definition of each predefined value for character\_code.

### 4.2.108.1.1 binary

binary: The document contains data in binary format.

### 4.2.108.1.2 IEC 61286

IEC 61286: The coded character set used to encode the document data according to IEC 61286.

### 4.2.108.1.3 ISO 646

ISO 646: The coded character set used to encode the document data according to ISO 646.

NOTE 2 The character set in ISO 646 is identical to the character set commonly known as ASCII.

### 4.2.108.1.4 ISO 3098-1

ISO 3098-1: The coded character set used to encode the document data is according to ISO 3098-1.

### 4.2.108.1.5 ISO 6937

IEC 61286: The coded character set used to encode the document data according to ISO/IEC 6937.

### 4.2.108.1.6 ISO 8859-1

ISO 8859-1: The coded character set used to encode the document data according to ISO 8859-1.

NOTE 3 The character set in ISO 8859-1 is identical to the character set commonly known as LATIN-1.

### 4.2.108.1.7 ISO 10646

ISO 10646: The coded character set used to encode the document data according to ISO/IEC 10646.



The character\_code need not be specified for a particular Document\_format\_property.

## 4.2.108.2 data\_format

The data\_format specifies the convention that was used to structure the information in the characterized object.

The predefined value of data\_format is one of the following:

- DXF;
- IGES;
- ISO 10303-203;
- ISO 10303-214;
- TIFF CCITT GR4.

NOTE 1 See 4.2.108.2.1 - 4.2.108.2.5 for the definition of each predefined value for data\_format.

### 4.2.108.2.1 DXF

DXF: The document contains data in Drawing (see 4.2.119) Exchange File format.

### 4.2.108.2.2 IGES

IGES: The document contains data in Initial Graphics Exchange Specification (see 4.2.323) format.

### 4.2.108.2.3 ISO 10303-203

ISO 10303-203: The document contains data in ISO 10303-203 format.

### 4.2.108.2.4 ISO 10303-214

ISO 10303-214: The document contains data in ISO 10303-214 format.

### 4.2.108.2.5 VDAFS

VDAFS: The document contains data in VDAFS format.

The data\_format need not be specified for a particular Document\_format\_property.

## 4.2.108.3 size\_format

The size\_format specifies the dimensions of a physical presentation of the object the size\_format applies to.

EXAMPLE 1 'ISO A0' is an example for the size\_format of a drawing that is stored digitally.

EXAMPLE 2 '0.2 x 0.4 x 0.4 meters' is an example for the size\_format of a wooden model.

See 4.3.997 for the application assertion.

## 4.2.109 Document\_location\_property

A Document\_location\_property specifies where a Document\_file (see 4.2.106) or a Document\_representation (see 4.2.110) can be found in a digital or physical data storage system. In the case where a Document\_location\_property is referred by a Document\_representation (see 4.2.110), the characteristics apply to all individual objects, whereas in the case it is referred by a Document\_file (see 4.2.106), the characteristics apply on an individual basis.

The data associated with a Document\_location\_property are the following:

— location\_name

### 4.2.109.1 location\_name

The location\_name specifies the location, where the object, that refers to the Document\_location\_property, can be found.

NOTE Multiple paths may be specified for a single object, e.g., a database in the context of an electronic vault.

EXAMPLE The linking mechanism of many operating systems allows references to a single object from various places in the file system without copying its contents.

## 4.2.110 Document\_representation

A Document\_representation is used to specify a specific issue of a document.

Each Document\_representation is either a Digital\_document (see 4.2.91), a Physical\_document (see 4.2.242), or a Physical\_model (see 4.2.244).

The data associated with a Document\_representation are the following:

- associated\_document\_version;
- common\_location;
- content;
- creation;
- document\_type;
- id;
- representation\_format;
- size.

#### **4.2.110.1 associated\_document\_version**

The associated\_document\_version specifies the version of the logical document that is being represented.

See 4.3.1003 for the application assertion.

#### **4.2.110.2 common\_location**

The common\_location describes the location of a document.

NOTE Different common locations represent alternative representations of the same physical location.

See 4.3.1001 for the application assertion.

#### **4.2.110.3 content**

The content specifies characteristics of the content of the Document\_representation.

The content need not be specified for a particular Document\_representation.

See 4.3.998 for the application assertion.

#### **4.2.110.4 creation**

The creation specifies further details of the creation of the Document\_representation.

The creation need not be specified for a particular Document\_representation.

See 4.3.999 for the application assertion.

#### **4.2.110.5 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Document\_representation.

The description need not be specified for a particular Document\_representation.

#### **4.2.110.6 id**

The id specifies the identifier of the Document\_representation.

#### **4.2.110.7 representation\_format**

The representation\_format specifies the format of the document represented by Document\_representation.

The representation\_format need not be specified for a particular Document\_representation.

See 4.3.1000 for the application assertion.

#### **4.2.110.8 size**

The size specifies the size of the represented document.

The size need not be specified for a particular Document\_representation.

See 4.3.1002 for the application assertion.

### **4.2.111 Document\_representation**

A Document\_representation (see 4.2.110) is used to specify a specific issue of a document.

The data associated with a Document\_representation (see 4.2.110) are the following:

- associated\_document\_version;
- description;
- document\_type;
- id.

#### **4.2.111.1 associated\_document\_version**

The associated\_document\_version specifies the Document\_version (see 4.2.114).

#### **4.2.111.2 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Document\_representation (see 4.2.110).

### 4.2.111.3 document\_type

The document\_type specifies the kind of the document represented by Document\_representation (see 4.2.110).

### 4.2.111.4 id

The id specifies the identifier of the Document\_representation (see 4.2.110).

## 4.2.112 Document\_size\_property

A Document\_size\_property specifies the size of a Document\_file (see 4.2.106) or of a Document\_representation (see 4.2.110) object. At least one of the optional attributes shall be specified for each instance of this object.

In the case where a Document\_size\_property is referred by a Document\_representation (see 4.2.110), the size information applies to the sum of all individual objects that are collected by this object, whereas in the case it is referred by a Document\_file (see 4.2.106), the size information is the one of the individual objects that is referenced.

The data associated with a Document\_size\_property are the following:

- file\_size;
- page\_count.

### 4.2.112.1 file\_size

The file\_size specifies the Value\_with\_unit (see 4.2.360) that represents the size of a digitally stored document. The file\_size shall only be applied in cases where the Document\_size\_property is referred by a Digital\_document (see 4.2.91) or a Document\_file (see 4.2.106).

EXAMPLE '15021 Bytes' and 'less than 500 Bytes' are examples for a file\_size.

The file\_size need not be specified for a particular Document\_size\_property.

See 4.3.1004 for the application assertion.

### 4.2.112.2 page\_count

The page\_count specifies the number of pages of the application object the Document\_size\_property referred by. The page\_count shall only be used in cases where the Document\_size\_property is referred by a Hardcopy (see 4.2.166) or a Physical\_document (see 4.2.242).

EXAMPLE 2 pages' and 'more than 1 page' are examples of a page\_count.

The page\_count need not be specified for a particular Document\_size\_property.

See 4.3.1005 for the application assertion.

## 4.2.113 Document\_structure

## **ISO 10303-212:2001(E)**

A Document\_structure is the relation between two Document\_representation (see 4.2.110) objects.

The data associated with an Document\_structure are the following:

- description;
- related;
- relating;
- relation\_type.

### **4.2.113.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Document\_structure.

The description need not be specified for a particular Document\_structure.

### **4.2.113.2 related**

The related specifies the second of the two Document\_representation (see 4.2.110) objects related by the Document\_structure.

See 4.3.1006 for the application assertion.

### **4.2.113.3 relating**

The relating specifies the first of the two Document\_representation (see 4.2.110) objects related by the Document\_structure.

See 4.3.1007 for the application assertion.

### **4.2.113.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- addition;
- copy;
- decomposition;
- derivation;
- peer;
- reference;
- sequence;
- substitution;
- translation.

NOTE See 4.2.113.4.1 - 4.2.113.4.8 for the definition of each predefined value for relation\_type.

#### 4.2.113.4.1 addition

addition: The Document\_structure specifies that the related document provides supplementary or collateral information with regard to the information provided by the relating document.

#### 4.2.113.4.2 copy

copy: The Document\_structure defines a relationship where the related Document\_representation (see 4.2.110) is a copy of the relating Document\_representation (see 4.2.110).

#### 4.2.113.4.3 decomposition

decomposition: The Document\_structure defines a relationship where the related Document\_representation (see 4.2.110) is one of potentially more sub documents of the relating Document\_representation (see 4.2.110).

#### 4.2.113.4.4 derivation

derivation: The Document\_structure defines a relationship where the related Document\_representation (see 4.2.110) is derived from the relating Document\_representation (see 4.2.110).

EXAMPLE Documentation prepared for specific target groups of end users is derived from a set of master documents. The information content is a subset of the information contained in the master document and the layout is tailored to the needs of the target group.

#### 4.2.113.4.5 reference

reference: The application object defines a relationship where the related document is referenced from the relating.

## ISO 10303-212:2001(E)

### 4.2.113.4.6 sequence

sequence: The Document\_structure defines a logical sequence where the related Document\_representation (see 4.2.110) comes after the relating Document\_representation (see 4.2.110) (e.g. a sequence of clauses).

### 4.2.113.4.7 substitution

substitution: The Document\_structure defines a relationship where the related Document\_representation (see 4.2.110) replaces the relating Document\_representation (see 4.2.110).

### 4.2.113.4.8 translation

translation: The Document\_structure specifies that the related document is generated through a translation process from the relating document.

## 4.2.114 Document\_type\_property

A Document\_type\_property specifies the kind of a Document\_file (see 4.2.106).

The data associated with an Document\_type\_property are the following:

- document\_type\_name;
- used\_classification\_system.

### 4.2.114.1 document\_type\_name

The document\_type\_name specifies the word or the group of words that describe the kind of object the characteristics are provided for. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- geometry;
- NC data;
- FE data;
- sample data;
- process plan;
- check plan;
- drawing.

NOTE See 4.2.114.1.1 - 4.2.114.1.8 for the definition of each predefined value for relation\_type.

#### 4.2.114.1.1 geometry

geometry: The document represents a shape model.



#### 4.2.114.1.2 NC data

NC data: The document represents numerical control data.

#### 4.2.114.1.3 FE data

FE data: The document represents numerical control data.

#### 4.2.114.1.4 sample data

sample data: The document represents measured data.

#### 4.2.114.1.5 process plan

process plan: The document represents process planning data.

#### 4.2.114.1.6 check plan

check plan: The document represents quality control planning data.

#### 4.2.114.1.7 drawing

drawing: The document represents a technical drawing.

### 4.2.114.2 used\_classification\_system

The used\_classification\_system specifies the Classification\_system (see 4.2.48) the document\_type\_ - name is defined in.

The used\_classification\_system need not be specified for a particular Document\_type\_property.

See 4.3.1008 for the application assertion.

### 4.2.115 Document\_version

A Document\_version specifies a particular variant of an Document (see 4.2.101).

NOTE Several versions for the same Document (see 4.2.101) may exist at one point in time. The information about valid and invalid versions is handled by the associated organizational data.

The data associated with an Document\_version are the following:

- associated\_document;
- description;
- id.

#### 4.2.115.1 associated\_document

The associated\_document specifies the Document (see 4.2.101) to which the Document\_version is assigned.

## ISO 10303-212:2001(E)

See 4.3.1009 for the application assertion.

### 4.2.115.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Document\_version.

The description need not be specified for a particular Document\_version.

### 4.2.115.3 id

The id specifies the identifier of the Document\_version.

### 4.2.116 Document\_version\_relationship

A Document\_version\_relationship is the relation between two Document\_version (see 4.2.114) objects.

The data associated with an Document\_version\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### 4.2.116.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Document\_version\_relationship.

The description need not be specified for a particular Document\_version\_relationship.

#### 4.2.116.2 related

The related specifies the second of the two Document\_version (see 4.2.114) objects related by the Document\_version\_relationship.

See 4.3.1010 for the application assertion.

#### 4.2.116.3 relating

The relating specifies the first of the two Document\_version (see 4.2.114) objects related by the Document\_version\_relationship.

See 4.3.1011 for the application assertion.

#### 4.2.116.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- derivation;
- hierarchy;
- sequence;
- supplied document.

NOTE See 4.2.116.4.1 - 4.2.116.4.4 for the definition of each predefined value for relation\_type.

##### 4.2.116.4.1 derivation

derivation: The Document\_version\_relationship defines a deriving relationship where the related Document\_version (see 4.2.114) is based on the relating Document\_version (see 4.2.114).

##### 4.2.116.4.2 hierarchy

hierarchy: The Document\_version\_relationship defines a hierarchical relationship where the related Document\_version (see 4.2.114) is a sub version of the relating Document\_version (see 4.2.114).

EXAMPLE Revision 1.1 and 1.2 of a document.

##### 4.2.116.4.3 sequence

sequence: The Document\_version\_relationship defines a succession of versions where the relating Document\_version (see 4.2.114) is the preceding version and the related Document\_version (see 4.2.114) is the following version. For a Document\_version (see 4.2.114), there shall be, at the most, one Document\_version\_relationship of this relation type as relating and, at most, one Document\_version\_relationship of this relation type as related.

##### 4.2.116.4.4 supplied document

supplied document: The Document\_version\_relationship defines a relationship where the related Document\_version (see 4.2.114) is an alias for the relating Document\_version (see 4.2.114). In this case the Document (see 4.2.101) objects associated with the two Document\_version (see 4.2.114) objects shall be different.

#### 4.2.117 Draughting\_annotation

A Draughting\_annotation is text and symbology applied to either a drawing sheet, drawing view, another piece of annotation, or a draughting model, for the purpose of communicating product data and drawing interpretation information.

Each Draughting\_annotation is either an Annotation\_element (see 4.2.15), an Annotation\_placed\_annotation (see 4.2.16), a Dimension (see 4.2.93), a Draughting\_callout (see 4.2.117), a Model\_

## ISO 10303-212:2001(E)

placed\_annotation (see 4.2.202), a Sheet\_placed\_annotation (see 4.2.308), or a View\_placed\_annotation (see 4.2.362).

### 4.2.118 Draughting\_callout

A Draughting\_callout is a type of Draughting\_annotation (see 4.2.116) that is a combination of text, annotation curves, and symbology that conveys information about a specific feature or area.

Each Draughting\_callout is either a Datum\_feature\_callout (see 4.2.83), a Datum\_target\_callout (see 4.2.84), or a Geometrical\_tolerance (see 4.2.160).

The data associated with a Draughting\_callout are the following:

- components.

#### 4.2.118.1 components

The components specifies the text, curves, and symbols that compose the draughting callout.

Each components may be one of the following: Annotation\_curve (see 4.2.14), Annotation\_symbol (see 4.2.20), or Text (see 4.2.342).

See 4.3.1012, 4.3.1013, and 4.3.1014 for the application assertions.

### 4.2.119 Draughting\_model

A Draughting\_model is the aggregation of all those elements that, taken as a whole, make up a graphical description of the product data of an electrotechnical system or of its constituents.

The data associated with a Draughting\_model are the following:

- coordinate\_space;
- elements;
- name.

#### 4.2.119.1 coordinate\_space

The coordinate\_space specifies the coordinate system that describes the two-dimensional space in which the content of the Draughting\_model is located in.

See 4.3.1015 for the application assertion.

#### 4.2.119.2 element

The element specifies the constituents of the Draughting\_model.

See 4.3.1016 for the application assertion.

#### 4.2.119.3 name

The name specifies the identifier of the Draughting\_model.

## 4.2.120 Drawing

A Drawing is the presentation of product data in a human-interpretable form wherein the physical and functional requirements for that product are presented in pictorial or textual form.

The data associated with a Drawing are the following:

- drawing\_specification;
- drawing\_type;
- id;
- language\_code;
- name;
- source;
- version\_id.

### 4.2.120.1 drawing\_specification

A drawing\_specification specifies the identification of the standard to which the drawing conforms. This standard specifies the presentation forms used in the drawing.

EXAMPLE A drawing\_specification can be ISO 129, Technical drawings – Dimensioning.

There shall be zero, one or more drawing\_specification for a Drawing.

### 4.2.120.2 drawing\_type

The drawing\_type specifies the category of the Drawing and may indicate the information content.

EXAMPLE A drawing\_type could be 'circuit diagram', 'terminal diagram', or 'arrangement drawing'.

### 4.2.120.3 extended\_designation

The extended\_designation specifies a structured label for the Drawing.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

The extended\_designation need not be specified for a particular Drawing.

See 4.3.1017 for the application assertion.

### 4.2.120.4 id

## ISO 10303-212:2001(E)

The id specifies the identifier of the Drawing.

### 4.2.120.5 language\_code

The language\_code specifies a language spoken by human beings to communicate with each other verbally or in written form. The language symbol given in ISO 639 shall be used.

There shall be zero, one or more language\_code for a Drawing.

### 4.2.120.6 name

The name specifies a speaking designation of the Drawing.

The name need not be specified for a particular Drawing.

### 4.2.120.7 source

The source specifies from where the drawing can be procured.

The source need not be specified for a particular Drawing.

### 4.2.120.8 version\_id

The version\_id specifies versioning information for the Drawing.

## 4.2.121 Drawing\_assignment

A Drawing\_assignment is a relationship that associates a Drawing (see 4.2.119) with an item.

The data associated with a Drawing\_assignment are the following:

- assigned\_drawing;
- is\_assigned\_to.

### 4.2.121.1 assigned\_drawing

The assigned\_drawing specifies the Drawing (see 4.2.119).

See 4.3.1037 for the application assertion.

### 4.2.121.2 is\_assigned\_to

The is\_assigned\_to specifies the item with which the Drawing (see 4.2.119) is associated.

Each is\_assigned\_to may be one of the following: Activity (see 4.2.1), Address (see 4.2.6), Approval (see 4.2.23), Approval\_status (see 4.2.25), Cable\_pull\_information (see 4.2.33), Class\_category - association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Classification\_attribute (see 4.2.47), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Connectivity\_definition (see 4.2.61), Contract (see 4.2.63), Data\_element (see 4.2.70), Data\_element\_definition (see 4.2.72), Data\_element\_specification (see 4.2.75), Design\_discipline\_item\_definition (see 4.2.86), Device (see

4.2.88), *Function\_definition* (see 4.2.145), *Function\_interface* (see 4.2.147), *Function\_unit* (see 4.2.148), *Function\_version* (see 4.2.150), *Functional\_connectivity\_definition* (see 4.2.152), *Functionality* (see 4.2.155), *General\_classification* (see 4.2.156), *Generic\_note* (see 4.2.159), *Interface* (see 4.2.170), *Interface\_port* (see 4.2.171), *Interface\_terminal* (see 4.2.174), *Item* (see 4.2.178), *Item\_identification* (see 4.2.180), *Item\_version* (see 4.2.182), *Location* (see 4.2.192), *Marking* (see 4.2.199), *Node* (see 4.2.208), *Notification* (see 4.2.213), *Object\_designation* (see 4.2.217), *Organization* (see 4.2.223), *Path* (see 4.2.232), *Path\_node* (see 4.2.233), *Person* (see 4.2.237), *Physical\_assembly\_relationship* (see 4.2.241), *Physical\_instance* (see 4.2.243), *Port* (see 4.2.247), *Process\_variable* (see 4.2.260), *Product\_class* (see 4.2.263), *Product\_identification* (see 4.2.268), *Project* (see 4.2.271), *Retention\_period* (see 4.2.289), *Route* (see 4.2.290), *Section* (see 4.2.296), *Section\_end* (see 4.2.297), *Section\_interface* (see 4.2.298), *Security\_classification* (see 4.2.301), *Security\_level* (see 4.2.302), *Signal* (see 4.2.309), *Signal\_value* (see 4.2.313), *Specification* (see 4.2.323), *Specification\_category* (see 4.2.324), *Specification\_expression* (see 4.2.326), *Specification\_inclusion* (see 4.2.327), *Technical\_system* (see 4.2.336), *Terminal* (see 4.2.338), *Work\_order* (see 4.2.364), or *Work\_request* (see 4.2.365).

See 4.3.1018, 4.3.1019, 4.3.1020, 4.3.1021, 4.3.1022, 4.3.1023, 4.3.1024, 4.3.1025, 4.3.1026, 4.3.1027, 4.3.1028, 4.3.1029, 4.3.1030, 4.3.1031, 4.3.1032, 4.3.1033, 4.3.1034, 4.3.1035, 4.3.1036, 4.3.1038, 4.3.1039, 4.3.1040, 4.3.1041, 4.3.1042, 4.3.1043, 4.3.1044, 4.3.1045, 4.3.1046, 4.3.1047, 4.3.1048, 4.3.1049, 4.3.1050, 4.3.1051, 4.3.1052, 4.3.1053, 4.3.1054, 4.3.1055, 4.3.1056, 4.3.1057, 4.3.1058, 4.3.1059, 4.3.1060, 4.3.1061, 4.3.1062, 4.3.1063, 4.3.1064, 4.3.1065, 4.3.1066, 4.3.1067, 4.3.1068, 4.3.1069, 4.3.1070, 4.3.1071, 4.3.1072, 4.3.1073, 4.3.1074, 4.3.1075, 4.3.1076, 4.3.1077, 4.3.1078, 4.3.1079, 4.3.1080, 4.3.1081, 4.3.1082, 4.3.1083, and 4.3.1084 for the application assertions.

## 4.2.122 *Drawing\_sequence*

A *Drawing\_sequence* is the relation between two *Drawing* (see 4.2.119) objects.

The data associated with a *Drawing\_sequence* are the following:

- *following\_version*;
- *preceding\_version*.

### 4.2.122.1 *following\_version*

The *following\_version* specifies the *Drawing* (see 4.2.119) that supersedes the preceding version of the *Drawing* (see 4.2.119).

See 4.3.1085 for the application assertion.

### 4.2.122.2 *preceding\_version*

The *preceding\_version* specifies the *Drawing* (see 4.2.119) that is superseded through the *Drawing* (see 4.2.119) associated by the '*following\_version*' attribute.

See 4.3.1086 for the application assertion.

### 4.2.123 Drawing\_sheet

A Drawing\_sheet is the logical division of a drawing into a two-dimensional area for the presentation of product data. These divisions correspond to sheet paper sizes for plotting. A Drawing\_sheet contains at least one Drawing\_view (see 4.2.125) or one Draughting\_annotation (see 4.2.116).

The data associated with a Drawing\_sheet are the following:

- associated\_drawing;
- coordinate\_space;
- extended\_designation;
- id;
- name;
- orientation;
- sheet\_number;
- size;
- version\_id.

#### 4.2.123.1 associated\_drawing

The associated\_drawing specifies the Drawing (see 4.2.119) that is valid for the Drawing\_sheet.

See 4.3.1089 for the application assertion.

#### 4.2.123.2 coordinate\_space

The coordinate\_space specifies the coordinate system that describes the two-dimensional space in which the content of the Drawing\_sheet is located.

See 4.3.1087 for the application assertion.

#### 4.2.123.3 extended\_designation

The extended\_designation specifies a structured label for the Drawing\_sheet.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'sheet\_number' attribute.

The extended\_designation need not be specified for a particular Drawing\_sheet.

See 4.3.1088 for the application assertion.

#### 4.2.123.4 name



The name specifies a speaking designation of the Drawing\_sheet.

The name need not be specified for a particular Drawing\_sheet.

#### **4.2.123.5 orientation**

The orientation specifies the alignment of the Drawing\_sheet.

The value of orientation is one of the following:

— landscape;

— portrait.

NOTE See 4.2.123.5.1 - 4.2.123.5.2 for the definition of each predefined value for orientation.

##### **4.2.123.5.1 landscape**

landscape: The Drawing\_sheet has a format where the width of the sheet is greater than the height.

##### **4.2.123.5.2 portrait**

portrait: The Drawing\_sheet has a format where the height of the sheet is greater than the width.

#### **4.2.123.6 sheet\_number**

The sheet\_number specifies the page number for a particular drawing sheet and its location in relation to other sheets of the drawing.

The sheet\_number need not be specified for a particular Drawing\_sheet.

#### **4.2.123.7 size**

The size specifies the physical size of the presentation area of the drawing sheet. This physical size corresponds to the physical size of a sheet of paper on which the drawing sheet can be placed.

See 4.3.1090 for the application assertion.

#### **4.2.123.8 version\_id**

The version\_id specifies versioning information for the Drawing\_sheet.

#### **4.2.124 Drawing\_sheet\_layout**

A Drawing\_sheet\_layout is a type of User\_defined\_symbol\_definition (see 4.2.355) that is the arrangement of a sheet of a drawing.

#### **4.2.125 Drawing\_sheet\_relationship**

A Drawing\_sheet\_relationship is the relation between two Drawing\_sheet (see 4.2.122) objects.

## ISO 10303-212:2001(E)

The data associated with an `Drawing_sheet_relationship` are the following:

- `description`;
- `related`;
- `relating`;
- `relation_type`.

### 4.2.125.1 `description`

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Drawing_sheet_relationship`.

The `description` need not be specified for a particular `Drawing_sheet_relationship`.

### 4.2.125.2 `related`

The `related` specifies the second of the two `Drawing_sheet` (see 4.2.122) objects related by the `Drawing_sheet_relationship`.

See 4.3.1091 for the application assertion.

### 4.2.125.3 `relating`

The `relating` specifies the first of the two `Drawing_sheet` (see 4.2.122) objects related by the `Drawing_sheet_relationship`.

See 4.3.1092 for the application assertion.

### 4.2.125.4 `relation_type`

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- `derivation`;
- `substitution`;
- `translation`;
- `version hierarchy`;
- `version sequence`.

NOTE See 4.2.125.4.1 - 4.2.125.4.5 for the definition of each predefined value for `relation_type`.

#### 4.2.125.4.1 `derivation`

derivation: The Drawing\_sheet\_relationship defines a deriving relationship where the related Drawing\_sheet (see 4.2.122) is based on the relating Drawing\_sheet (see 4.2.122).

#### 4.2.125.4.2 substitution

substitution: The Drawing\_sheet\_relationship defines a relationship where the related Drawing\_sheet (see 4.2.122) replaces the relating Drawing\_sheet (see 4.2.122).

#### 4.2.125.4.3 translation

translation: The Drawing\_sheet\_relationship defines a relationship where the related Drawing\_sheet (see 4.2.122) is a transcription into another language of the relating Drawing\_sheet (see 4.2.122).

#### 4.2.125.4.4 version hierarchy

version hierarchy: The Drawing\_sheet\_relationship defines a hierarchical relationship where the related Drawing\_sheet (see 4.2.122) is a subversion of the relating Drawing\_sheet (see 4.2.122).

EXAMPLE Revision 1.1 and 1.2 of a drawing.

#### 4.2.125.4.5 version sequence

version sequence: The Drawing\_sheet\_relationship defines a succession of versions where the relating Drawing\_sheet (see 4.2.122) is the preceding version and the related Drawing\_sheet (see 4.2.122) is the following version. For a Drawing\_sheet (see 4.2.122) there shall be, at the most, one Drawing\_sheet\_relationship of this relation type as relating and, at most, one Drawing\_sheet\_relationship of this relation type as related.

### 4.2.126 Drawing\_view

A Drawing\_view is the set of instructions for producing a two-dimensional planar projection of a Draughting\_model (see 4.2.118) from a specified position within its coordinate system.

## ISO 10303-212:2001(E)

NOTE A `Drawing_view` may be placed multiple times on a drawing, but each is a separate `Drawing_view`. The placement within a particular `Drawing_sheet` (see 4.2.122) makes each `Drawing_view` unique to that `Drawing_sheet` (see 4.2.122).

The data associated with a `Drawing_view` are the following:

- `containing_sheet`;
- `coordinate_space`;
- `id`;
- `position`;
- `rotation`.

### 4.2.126.1 `containing_sheet`

The `containing_sheet` specifies the `Drawing_sheet` (see 4.2.122) object in which the `Drawing_view` is placed.

See 4.3.1094 for the application assertion.

### 4.2.126.2 `coordinate_space`

The `coordinate_space` specifies the `Cartesian_coordinate_space_2d` (see 4.2.34) in which the `Drawing_view` has been defined.

See 4.3.1093 for the application assertion.

### 4.2.126.3 `id`

The `id` specifies the identifier of the `Drawing_view`.

The `id` need not be specified for a particular `Drawing_view`.

### 4.2.126.4 `position`

The `position` specifies the location of the origin of the coordinate system of the `Drawing_view` relative to the origin of the coordinate system of the drawing sheet where it is placed.

See 4.3.1095 for the application assertion.

### 4.2.126.5 `rotation`

The `rotation` specifies the angle, measured counter-clockwise, between the horizontal axis of the coordinate system of the `Drawing_view` and the horizontal axis of the coordinate system of the `Drawing_sheet` (see 4.2.122) where it is placed.

### 4.2.127 `Duration`

A Duration is the definition of a period of time.

The data associated with an Effectivity (see 4.2.127) are the following:

- time;
- time\_unit.

#### **4.2.127.1 time**

The time specifies the extend of the Duration.

#### **4.2.127.2 time\_unit**

The time\_unit specifies the unit in which the time is specified.

### **4.2.128 Effectivity**

An Effectivity is the identification of the valid use of an aspect of product data tracked by date.

The data associated with an Effectivity are the following:

- concerned\_organization;
- description;
- effectivity\_context;
- end\_definition;
- id;
- period;
- start\_definition;
- version\_id.

#### **4.2.128.1 concerned\_organization**

The concerned\_organization specifies the Organization (see 4.2.223) for which the Effectivity is valid.

**EXAMPLE** The Effectivity of the same item may be different in the various production sites of the manufacturer.

See 4.3.1101 for the application assertion.

#### **4.2.128.2 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Effectivity.

## ISO 10303-212:2001(E)

The description need not be specified for a particular Effectivity.

### 4.2.128.3 effectivity\_context

The effectivity\_context specifies the life-cycle stage for which the Effectivity is valid.

EXAMPLE The effectivity\_context is 'prototype building' in order to express that the Effectivity is valid for this life-cycle stage of the concerned item.

The effectivity\_context need not be specified for a particular Effectivity.

### 4.2.128.4 end\_definition

The end\_definition specifies the date, event, or period of time after the start\_definition or after an event when the identified item is no longer effective.

The end\_definition need not be specified for a particular Effectivity.

Each end\_definition may be one of the following: Date\_time (see 4.2.79) or Event\_reference (see 4.2.130).

See 4.3.1096 and 4.3.1099 for the application assertions.

### 4.2.128.5 id

The id specifies the identifier of the Effectivity.

The id need not be specified for a particular Effectivity.

### 4.2.128.6 period

The period specifies the period of time in which the Effectivity is valid, starting at the point in time specified by either 'start\_definition' or 'end\_definition'.

The period need not be specified for a particular Effectivity.

See 4.3.1098 for the application assertion.

### 4.2.128.7 start\_definition

The start\_definition specifies the date or the event when the identified item becomes effective.

The start\_definition need not be specified for a particular Effectivity.

Each start\_definition may be one of the following: Date\_time (see 4.2.79) or Event\_reference (see 4.2.130).

See 4.3.1097 and 4.3.1100 for the application assertions.

### 4.2.128.8 version\_id

The version\_id specifies versioning information for the Effectivity.

The version\_id need not be specified for a particular Effectivity.

## 4.2.129 Effectivity\_assignment

An Effectivity\_assignment associates an Effectivity (see 4.2.127) with an item, where the effectivity of the item is controlled by the associated Effectivity (see 4.2.127).

The data associated with an Effectivity\_assignment are the following:

- assigned\_effectivity;
- effectivity\_indication;
- is\_applied\_to;
- role.

### 4.2.129.1 assigned\_effectivity

The assigned\_effectivity specifies the Effectivity (see 4.2.127) object.

See 4.3.1136 for the application assertion.

### 4.2.129.2 effectivity\_indication

The effectivity\_indication specifies whether the Effectivity\_assignment takes effect. A value of 'true' indicates that the Effectivity\_assignment is effective, a value of 'false' that the Effectivity\_assignment is ineffective.

### 4.2.129.3 is\_applied\_to

The is\_applied\_to specifies the item with which the Effectivity (see 4.2.127) object is associated.

Each is\_applied\_to may be one of the following: Activity\_relationship (see 4.2.5), Alternate\_item\_relationship (see 4.2.11), Assembly\_component\_relationship (see 4.2.26), Assembly\_substitute\_relationship (see 4.2.28), Cable\_pull\_information (see 4.2.33), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see 4.2.124), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see

## ISO 10303-212:2001(E)

4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version - relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node - relationship (see 4.2.234), Path\_relationship (see 4.2.235), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Requirement (see 4.2.285), Route (see 4.2.290), Route\_relationship (see 4.2.291), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), or Terminal (see 4.2.338).

See 4.3.1102, 4.3.1103, 4.3.1104, 4.3.1105, 4.3.1106, 4.3.1107, 4.3.1108, 4.3.1109, 4.3.1110, 4.3.1111, 4.3.1112, 4.3.1113, 4.3.1114, 4.3.1115, 4.3.1116, 4.3.1117, 4.3.1118, 4.3.1119, 4.3.1120, 4.3.1121, 4.3.1122, 4.3.1123, 4.3.1124, 4.3.1125, 4.3.1126, 4.3.1127, 4.3.1128, 4.3.1129, 4.3.1130, 4.3.1131, 4.3.1132, 4.3.1133, 4.3.1134, 4.3.1135, 4.3.1137, 4.3.1138, 4.3.1139, 4.3.1140, 4.3.1141, 4.3.1142, 4.3.1143, 4.3.1144, 4.3.1145, 4.3.1146, 4.3.1147, 4.3.1148, 4.3.1149, 4.3.1150, 4.3.1151, 4.3.1152, 4.3.1153, 4.3.1154, 4.3.1155, 4.3.1156, 4.3.1157, 4.3.1158, 4.3.1159, 4.3.1160, 4.3.1161, 4.3.1162, 4.3.1163, 4.3.1164, 4.3.1165, 4.3.1166, 4.3.1167, 4.3.1168, 4.3.1169, 4.3.1170, 4.3.1171, 4.3.1172, 4.3.1173, 4.3.1174, 4.3.1175, 4.3.1176, 4.3.1177, 4.3.1178, 4.3.1179, 4.3.1180, 4.3.1181, 4.3.1182, 4.3.1183, 4.3.1184, 4.3.1185, 4.3.1186, 4.3.1187, 4.3.1188, 4.3.1189, 4.3.1190, and 4.3.1191 for the application assertions.

### 4.2.129.4 role

The role specifies the relationship between the Effectivity (see 4.2.127) and the item that has an effectivity assigned to it. The value is either user defined or predefined.

The predefined value of role is one of the following:

- actual;
- planned;
- required.

NOTE See 4.2.129.4.1 - 4.2.129.4.3 for the definition of each predefined value for relation\_type.

#### 4.2.129.4.1 actual

actual: The actual period during which the Effectivity (see 4.2.127) lasted.

#### 4.2.129.4.2 planned

planned: The period associated with the Effectivity (see 4.2.127) defines a planned period of time during which the associated object is or was supposed to be effective.



**4.2.129.4.3 required**

required: The associated object must be kept effective for this period.

**4.2.130 Effectivity\_relationship**

An Effectivity\_relationship is the relation between two Effectivity (see 4.2.127) objects.

NOTE Sometimes the effectivity is not dependent on particular dates but on the effectivity of other items. In this case, the dates are not instantiated, and an Effectivity (see 4.2.127) relationship to the reference Effectivity (see 4.2.127) exists.

The data associated with an Effectivity\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

**4.2.130.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Effectivity\_relationship.

The description need not be specified for a particular Effectivity\_relationship.

**4.2.130.2 related**

The related specifies the second of the two Effectivity (see 4.2.127) objects related by the Effectivity\_relationship.

See 4.3.1192 for the application assertion.

**4.2.130.3 relating**

The relating specifies the first of the two Effectivity (see 4.2.127) objects related by the Effectivity\_relationship.

See 4.3.1193 for the application assertion.

**4.2.130.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- `constraint`;
- `inheritance`.

NOTE See 4.2.130.4.1 - 4.2.130.4.2 for the definition of each predefined value for `relation_type`.

### 4.2.130.4.1 `constraint`

`constraint`: The time period between the start and end definition of the related Effectivity (see 4.2.127) shall be within the time period of the relating Effectivity (see 4.2.127).

### 4.2.130.4.2 `inheritance`

`inheritance`: The related Effectivity (see 4.2.127) shall not have a 'start definition' and 'end definition' specified but inherits the effectivity dates from the relating Effectivity (see 4.2.127).

## 4.2.131 `Event_reference`

An `Event_reference` is the definition of a point in time established with respect to an event.

The data associated with an `Event_reference` are the following:

- `description`;
- `event_context`;
- `event_type`;
- `offset`.

### 4.2.131.1 `description`

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Event_reference`.

The `description` need not be specified for a particular `Event_reference`.

### 4.2.131.2 `event_context`

The `event_context` specifies the piece of product data the `Event_reference` refers to.

EXAMPLE In the case of an `Event_reference` with event type 'start of production', the event context would refer to the `Item_version` (see 4.2.182) that is subject to production.

The `event_context` need not be specified for a particular `Event_reference`.

Each `event_context` may be one of the following: `Activity` (see 4.2.1), `Activity_element` (see 4.2.2), `Activity_method_assignment` (see 4.2.4), `Activity_relationship` (see 4.2.5), `Alternate_item_relationship` (see 4.2.11), `Approval_status` (see 4.2.25), `Assembly_component_relationship` (see 4.2.26), `Assembly_substitute_relationship` (see 4.2.28), `Cable_pull_information` (see 4.2.33),

Certification (see 4.2.38), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_association (see 4.2.46), Classification\_attribute (see 4.2.47), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_allocation (see 4.2.60), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Contract (see 4.2.63), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Data\_element\_specification (see 4.2.75), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see 4.2.124), Free\_segment (see 4.2.144), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), Functionality (see 4.2.155), General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Organization\_relationship (see 4.2.225), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Person\_in\_organization (see 4.2.238), Person\_in\_organization\_relationship (see 4.2.239), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Port\_association (see 4.2.249), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Requirement\_document\_assignment (see 4.2.287), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.1194, 4.3.1195, 4.3.1196, 4.3.1197, 4.3.1198, 4.3.1199, 4.3.1200, 4.3.1201, 4.3.1202, 4.3.1203, 4.3.1204, 4.3.1205, 4.3.1206, 4.3.1207, 4.3.1208, 4.3.1209, 4.3.1210, 4.3.1211, 4.3.1212, 4.3.1213, 4.3.1214, 4.3.1215, 4.3.1216, 4.3.1217, 4.3.1218, 4.3.1219, 4.3.1220, 4.3.1221, 4.3.1222, 4.3.1223, 4.3.1224, 4.3.1225, 4.3.1226, 4.3.1227, 4.3.1228, 4.3.1229, 4.3.1230, 4.3.1231, 4.3.1232, 4.3.1233, 4.3.1234, 4.3.1235, 4.3.1236, 4.3.1237, 4.3.1239, 4.3.1240, 4.3.1241, 4.3.1242, 4.3.1243, 4.3.1244, 4.3.1245, 4.3.1246, 4.3.1247, 4.3.1248, 4.3.1249, 4.3.1250, 4.3.1251, 4.3.1252, 4.3.1253, 4.3.1254, 4.3.1255, 4.3.1256, 4.3.1257, 4.3.1258, 4.3.1259, 4.3.1260, 4.3.1261, 4.3.1262, 4.3.1263, 4.3.1264, 4.3.1265, 4.3.1266, 4.3.1267, 4.3.1268, 4.3.1269, 4.3.1270, 4.3.1271, 4.3.1272, 4.3.1273, 4.3.1274, 4.3.1275, 4.3.1276, 4.3.1277, 4.3.1278, 4.3.1279, 4.3.1280, 4.3.1281, 4.3.1282, 4.3.1283,

## ISO 10303-212:2001(E)

4.3.1284, 4.3.1285, 4.3.1286, 4.3.1287, 4.3.1288, 4.3.1289, 4.3.1290, 4.3.1291, 4.3.1292, 4.3.1293, 4.3.1294, 4.3.1295, 4.3.1296, 4.3.1297, 4.3.1298, 4.3.1299, 4.3.1300, 4.3.1301, 4.3.1302, 4.3.1303, 4.3.1304, 4.3.1305, 4.3.1306, 4.3.1307, 4.3.1308, 4.3.1309, 4.3.1310, 4.3.1311, and 4.3.1312 for the application assertions.

### 4.2.131.3 event\_type

The event\_type specifies the kind of event that serves as reference.

EXAMPLE The cases 'start of production' or 'end of production' are examples for the event\_type.

### 4.2.131.4 offset

The offset specifies the amount of time before or after the defined event that shall be used to calculate the actual point in time.

The offset need not be specified for a particular Event\_reference.

See 4.3.1238 for the application assertion.

### 4.2.132 External\_file\_id\_and\_location

An External\_file\_id\_and\_location specifies the location of an file in an external storage system.

The data associated with an External\_file\_id\_and\_location are the following:

- external\_id;
- location.

#### 4.2.132.1 external\_id

The external\_id specifies the identifier of an document in the external location.

#### 4.2.132.2 location

The location specifies the location of the Document\_file (see 4.2.106) in the external storage system.

See 4.3.1313 for the application assertion.

### 4.2.133 External\_library\_reference

An External\_library\_reference is a mechanism to refer to an entry in an external library other than ISO 13584 or IEC 61360.

The data associated with an External\_library\_reference are the following:

- description;
- external\_id;
- library\_type.

#### **4.2.133.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the External\_library\_reference.

The description need not be specified for a particular External\_library\_reference.

#### **4.2.133.2 external\_id**

The external\_id specifies the unique identifier of the referenced entry in the external library.

#### **4.2.133.3 library\_type**

The library\_type specifies the type of library that is used.

### **4.2.134 Externally\_defined\_hatching**

An Externally\_defined\_hatching is a type of Fill\_area\_appearance (see 4.2.140) that has a specific physical appearance defining the hatching and is found in a known source. This known source is agreed to by all parties involved in the exchange of the drawings on which the hatching pattern appears. The specific appearance is referred to as a hatch pattern and consists of a curve appearance and the angle and displacement values necessary to define a single, uniformly spaced geometric pattern. An Externally\_defined\_hatching shall include the specification of the curve appearance, the angle of the curves in the pattern relative to the horizontal axis of the coordinate system into which it is placed, and the displacement between adjacent curves in the pattern.

The data associated with an Externally\_defined\_hatching are the following:

- hatching\_name;
- hatching\_reference.

#### **4.2.134.1 hatching\_name**

The hatching\_name specifies the identification of a particular hatching pattern within the known source.

#### **4.2.134.2 hatching\_reference**

The hatching\_reference specifies the known source that contains a set of patterns from which the hatching pattern is selected.

### **4.2.135 Externally\_defined\_line\_font**

## ISO 10303-212:2001(E)

An Externally\_defined\_line\_font is a type of Line\_font (see 4.2.189) that has a specific physical appearance defining the line font and is found in a known source. This known source is agreed to by all parties involved in the exchange of the drawings on which the line font appears. An Externally\_defined\_line\_font shall include a set of values that represent the length of the visible and invisible segments of the line font. The set of values is sufficient to define all elements that constitute a single portion of the curve. This portion is then repeated over the length of the curve.

The data associated with an Externally\_defined\_line\_font are the following:

- font\_id;
- font\_reference.

### 4.2.135.1 font\_id

The font\_id specifies the identification of a particular line font within the known source.

### 4.2.135.2 font\_reference

The font\_reference specifies the known source that contains a set of line fonts from which the line font is selected.

### 4.2.136 Externally\_defined\_symbol

An Externally\_defined\_symbol is a type of Annotation\_symbol (see 4.2.20) that has a specific physical appearance defining the symbol and is found in a known source. This known source is agreed to by all parties involved in the exchange of the drawings on which the symbol appears. An Externally\_defined\_symbol shall include the specification of all the constituent components of the symbol, their size, and relative locations.

EXAMPLE IEC 60617 and IEC 81714 specify the layout of graphical symbols for use in diagrams.

The data associated with an Externally\_defined\_symbol are the following:

- symbol\_name;
- symbol\_reference.

#### 4.2.136.1 symbol\_name

The symbol\_name specifies the identification of a particular symbol within the known source.

#### 4.2.136.2 symbol\_reference

The symbol\_reference specifies the known source that contains a set of symbols from which the symbol is selected.

### 4.2.137 Externally\_defined\_text\_font

An Externally\_defined\_text\_font is a type of Text\_font (see 4.2.344) that has a specific physical appearance defining the text font and is found in a known source. This known source is agreed to by

all parties involved in the exchange of the drawings on which the text font appears. An `Externally_defined_text_font` shall include the specification of the physical form of the characters of the font.

The data associated with an `Externally_defined_text_font` are the following:

- `font_id`;
- `font_reference`.

#### **4.2.137.1 font\_id**

The `font_id` specifies the identification of a particular text font within the known source.

#### **4.2.137.2 font\_reference**

The `font_reference` specifies the known source that contains a set of text fonts from which the text font is selected.

### **4.2.138 Externally\_defined\_tile**

An `Externally_defined_tile` is a type of `Tile` (see 4.2.346) that has a specific physical appearance defining the tile and is found in a known source. This known source is agreed to by all parties involved in the exchange of the drawings on which the tile appears.

The data associated with an `Externally_defined_tile` are the following:

- `tile_name`;
- `tile_reference`.

#### **4.2.138.1 tile\_name**

The `tile_name` specifies the identification of a particular tile within the known source.

#### **4.2.138.2 tile\_reference**

The `tile_reference` specifies the known source that contains a set of tiles from which the tile is selected.

### **4.2.139 Externally\_defined\_tiling**

An `Externally_defined_tiling` is a type of `Fill_area_appearance` (see 4.2.140) that has a specific physical appearance defining the tiling pattern and is found in a known source. This known source is agreed to by all parties involved in the exchange of the drawings on which the tiling appears. An `Externally_defined_tiling` shall include the repeat vectors used to define the relative positioning of tiles, the angle of the horizontal axis of the tile relative to the horizontal axis of the coordinate system into which it is placed, and the scale of the tile as presented to the tile as defined.

## ISO 10303-212:2001(E)

The data associated with an Externally\_defined\_tiling are the following:

- tiling\_name;
- tiling\_reference.

### 4.2.139.1 tiling\_name

The tiling\_name specifies the identification of a particular tiling patterns within the known source.

### 4.2.139.2 tiling\_reference

The tiling\_reference specifies the known source that contains a set of pattern from which the tiling pattern is selected.

### 4.2.140 Fill\_area

A Fill\_area is a type of Annotation\_element (see 4.2.15) that is a bounded area containing colouring, hatching, or tiling that indicate its extent and content. A Fill\_area communicates some aspect of a physical part characteristic, distinguishes some aspect of a physical part from its surroundings, is part of another piece of annotation, or is used as an annotation by itself.

NOTE Fill areas are derived from geometric elements, annotation curves, or a combination of both. The two types of curves can be combined if the geometric elements do not result in a closed boundary necessary for the confinement of the filled area. Only those annotation curves used for geometric construction can be included in the boundary of a fill area.

The data associated with a Fill\_area are the following:

- assigned\_appearance;
- bounds;
- reference\_point.

#### 4.2.140.1 assigned\_appearance

The assigned\_appearance specifies the definition of the appearance characteristics of the Fill\_area.

See 4.3.1314 for the application assertion.

#### 4.2.140.2 boundary

The boundary specifies the outline of an Fill\_area.

See 4.3.1315 for the application assertion.

#### 4.2.140.3 reference\_point

The reference\_point specifies a point within the fill area used in the placement and initiation of the fill area appearance. The reference\_point establishes a point through which a line of a hatching pattern passes or at which the origin of a tile is located. The reference\_point also establishes the point



at which the first visible segment of a line font used as the curve appearance for a hatching pattern starts.

The reference\_point need not be specified for a particular Fill\_area.

See 4.3.1316 for the application assertion.

#### **4.2.141 Fill\_area\_appearance**

A Fill\_area\_appearance is a type of Appearance (see 4.2.21) that governs the visual presentation of a fill area.

Each Fill\_area\_appearance is either an Externally\_defined\_hatching (see 4.2.133), an Externally\_defined\_tiling (see 4.2.138), a Solid\_fill\_area (see 4.2.317), a User\_defined\_hatching (see 4.2.352), or a User\_defined\_tiling (see 4.2.357).

The data associated with a Fill\_area\_appearance are the following:

- draughting\_role.

##### **4.2.141.1 draughting\_role**

The draughting\_role specifies the purpose within draughting for a particular fill area appearance.

The draughting\_role need not be specified for a particular Fill\_area\_appearance.

#### **4.2.142 Fill\_area\_boundary**

A Fill\_area\_boundary is an annotation curve in the same coordinate space that defines the limits of a fill area. The curve is closed and not self intersecting. In a three-dimensional fill area, the curve either forms a closed curve on a planar surface or is coincident with the boundary of a closed surface.

NOTE The curves composing the Fill\_area\_boundary are derived from, and coincide with, the geometric curves or surfaces and the annotation curves that define the extent of the fill area.

The data associated with a Fill\_area\_boundary are the following:

- defining\_curve.

##### **4.2.142.1 defining\_curve**

The defining\_curve specifies the annotation curve that defines the boundary of the fill area.

See 4.3.1317 for the application assertion.

#### **4.2.143 Final\_solution**

A Final\_solution is a type of Alternative\_solution (see 4.2.12) that is the specification of a set of items that fulfil the same functional requirements as the neutral base element and have additional characteristics.

EXAMPLE The neutral parts are parts without paint; the final parts are the parts with paint.

## ISO 10303-212:2001(E)

The data associated with a `Final_solution` are the following:

- `final_specification`;
- `final_status`.

### 4.2.143.1 `final_specification`

The `final_specification` specifies the means of finalization that is applied to the neutral part.

EXAMPLE The `final_specification` can be the `Design_discipline_item_definition` (see 4.2.86) of a certain kind of paint.

NOTE The `Design_discipline_item_definition` (see 4.2.86) that is referenced here is not a view of the equipment that results from the finalization. The resulting, coloured devices can be identified by the `Device` (see 4.2.88) objects that are associated with the `Final_solution`.

Each `final_specification` may be one of the following: `Descriptive_specification` (see 4.2.85), `Design_discipline_item_definition` (see 4.2.86), `Function_definition` (see 4.2.145), or `Physical_instance` (see 4.2.243).

See 4.3.1318, 4.3.1319, 4.3.1320, and 4.3.1321 for the application assertions.

### 4.2.143.2 `final_status`

The `final_status` specifies the level of completion between the neutral part and the final part.

EXAMPLE The status information 'final for shipping overseas', 'final for transport by truck', or 'final for sale' are examples for the `final_status`.

### 4.2.144 `Format_of_value`

A `Format_of_value` is the specification of the syntactical format of a value. If present, the '`value_of_single_value`' attribute of the corresponding `Single_value` (see 4.2.316) object shall contain only data that conform with the `Format_of_value` object.

The data associated with a `Format_of_value` are the following:

- `associated_definition`;
- `default_language_specification`;
- `source_document`;
- `value_format`.

#### 4.2.144.1 `associated_definition`

The `associated_definition` specifies the appropriate `Data_element_definition` (see 4.2.72).

See 4.3.1322 for the application assertion.

#### 4.2.144.2 default\_language\_specification

The default\_language\_specification specifies a language spoken by human beings to communicate with each other verbally or in written form. This language is the preselected language used in the associated String\_value (see 4.2.331) object. The language symbol given in ISO 639 shall be used.

The default\_language\_specification need not be specified for a particular Format\_of\_value.

See 4.3.1323 for the application assertion.

#### 4.2.144.3 source\_document

The source\_document specifies the identifier of the document in which the value\_format is specified. The value is either user defined or predefined.

The predefined value of source\_document is the following:

— iec 61360.

NOTE See 4.2.144.3.1 for the definition of each predefined value for source\_document.

##### 4.2.144.3.1 iec 61360

iec 61360: The value\_format is in accordance with IEC 61360.

#### 4.2.144.4 value\_format

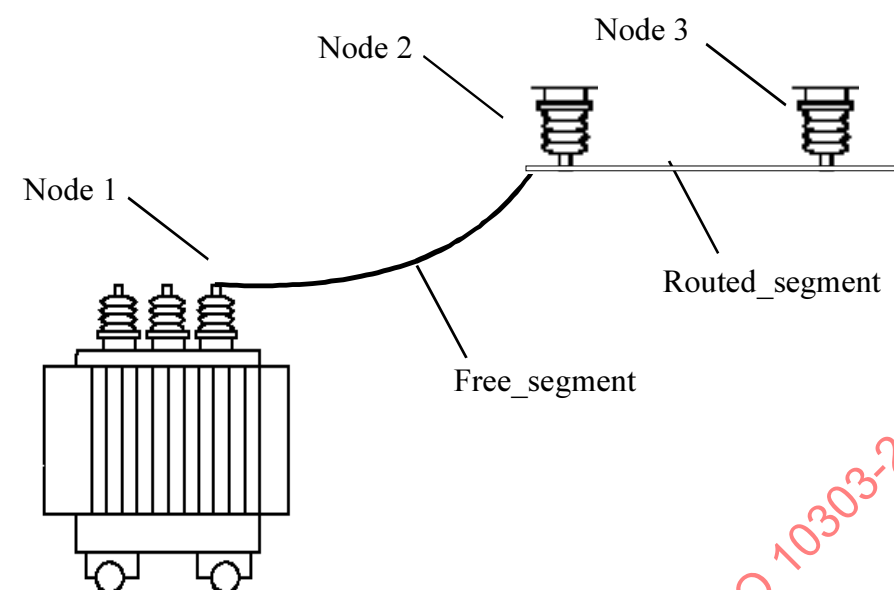
The value\_format specifies the syntactical format of the value that is associated to the appropriate Data\_element\_definition (see 4.2.72) object.

#### 4.2.145 Free\_segment

A Free\_segment is a portion of a specific path that does not have a well-defined course.

EXAMPLE 1 Figure 13 shows a transformer that is moveable on rails. The cable that feeds the transformer is mounted on the ceiling. The geometry of the section between the last isolator and the transformer changes depending on the position of the transformer. In this case, there will be two Node (see 4.2.208) objects in the ends\_at attribute of the Free\_segment entity.

EXAMPLE 2 If in Figure 13 the transformer were not yet installed, the last well defined node of the cable is Node (see 4.2.208) 2. In this case, there will be only one Node (see 4.2.208) object in the ends\_at attribute of the Free\_segment entity. Such cases occur throughout the installation of a system if some design sections are finished earlier than others.



**Figure 13 - Power transformer fed through an overhead cable**

The data associated with a Free\_segment are the following:

- description;
- ends\_at;
- id.

#### **4.2.145.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Free\_segment.

The description need not be specified for a particular Free\_segment.

#### **4.2.145.2 ends\_at**

The ends\_at specifies the Node (see 4.2.208) objects that delimits the Free\_segment.

See 4.3.1324 for the application assertion.

#### **4.2.145.3 id**

The id specifies the identifier of the Free\_segment.

#### **4.2.146 Function\_definition**

A Function\_definition is the characterization of a Function\_version (see 4.2.150) in a particular application context.

The data associated with a Function\_definition are the following:

- additional\_context;
- id;
- initial\_context;
- name;
- version.

#### **4.2.146.1 additional\_context**

The additional\_context specifies the set of Application\_context (see 4.2.22) objects in which this view of the Function\_version (see 4.2.150) is also relevant. The additional\_context shall not contain the Application\_context (see 4.2.22) that is referenced as the 'initial\_context'.

See 4.3.1325 for the application assertion.

#### **4.2.146.2 id**

The id specifies the identifier of the Function\_definition.

#### **4.2.146.3 initial\_context**

The initial\_context specifies the Application\_context (see 4.2.22) in which this view of the Function\_version (see 4.2.150) has been designed primarily.

See 4.3.1326 for the application assertion.

#### **4.2.146.4 name**

The name specifies a speaking designation of the Function\_definition.

The name need not be specified for a particular Function\_definition.

#### **4.2.146.5 version**

The version specifies the Function\_version (see 4.2.150) object to which the Function\_definition relates.

See 4.3.1327 for the application assertion.

### **4.2.147 Function\_definition\_relationship**

A Function\_definition\_relationship is the relation between two Function\_definition (see 4.2.145) objects.

## ISO 10303-212:2001(E)

The data associated with an `Function_definition_relationship` are the following:

- `description`;
- `related`;
- `relating`;
- `relation_type`.

### 4.2.147.1 `description`

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Function_definition_relationship`.

The `description` need not be specified for a particular `Function_definition_relationship`.

### 4.2.147.2 `related`

The `related` specifies the second of the two `Function_definition` (see 4.2.145) objects related by the `Function_definition_relationship`.

See 4.3.1328 for the application assertion.

### 4.2.147.3 `relating`

The `relating` specifies the first of the two `Function_definition` (see 4.2.145) objects related by the `Function_definition_relationship`.

See 4.3.1329 for the application assertion.

### 4.2.147.4 `relation_type`

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- `alternate`;
- `derivation`;
- `substitution`.

NOTE 1 See 4.2.147.4.1 - 4.2.147.4.3 for the definition of each predefined value for `relation_type`.

#### 4.2.147.4.1 `alternate`

`alternate`: The `Function_definition_relationship` defines a relationship where the related `Function_definition` (see 4.2.145) is a possible substitute to the relating `Function_definition` (see 4.2.145).

NOTE 2 This concept refers to the possibility to replace the related Function\_definition (see 4.2.145). The actual replacement is addressed by 'substitution'.

#### 4.2.147.4.2 derivation

derivation: The Function\_definition\_relationship defines a deriving relationship where the related Function\_definition (see 4.2.145) is based on the relating Function\_definition (see 4.2.145).

#### 4.2.147.4.3 substitution

substitution: The Function\_definition\_relationship defines a relationship where the related Function\_definition (see 4.2.145) replaces the relating Function\_definition (see 4.2.145).

### 4.2.148 Function\_interface

A Function\_interface specifies Interface\_port (see 4.2.171) objects that characterize the use or the intended purpose of a functional module.

NOTE The Function\_interface can provide a possible selection criterion if a given functional module needs to be substituted by a different one. Possible selection criteria may be assigned as Data\_element (see 4.2.70) objects to Function\_interface.

The data associated with a Function\_interface are the following:

- description;
- external\_access;
- id;
- interface\_of.

#### 4.2.148.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Function\_interface.

The description need not be specified for a particular Function\_interface.

#### 4.2.148.2 external\_access

The external\_access specifies the interface by assigning Interface\_port (see 4.2.171) objects to the Function\_interface.

See 4.3.1331 for the application assertion.

#### 4.2.148.3 id

The id specifies the identifier of the Function\_interface.

#### 4.2.148.4 interface\_of

## ISO 10303-212:2001(E)

The interface\_of specifies the associated Function\_definition (see 4.2.145).

See 4.3.1330 for the application assertion.

### 4.2.149 Function\_unit

A Function\_unit is a type of Product\_constituent (see 4.2.266) that is an occurrence of a Function\_definition (see 4.2.145) object. A Function\_unit may be instantiated more than once, because each instance is an individual occurrence of the functional item that is characterized by the Function\_definition (see 4.2.145).

EXAMPLE 1 In a specific technical system, the function 'amplifier' is defined once. This Function\_definition (see 4.2.145) carries all the information defining the amplifier (e.g., its ports) that is independent from its usage. Additionally, two Function\_unit objects for this amplifier exist because two equal amplifiers are used within this particular circuitry. Each of these instances may be connected individually.

Each Function\_unit is either a Single\_function\_unit (see 4.2.315) or a Specified\_function\_unit (see 4.2.329).

The data associated with a Function\_unit are the following:

- description;
- extended\_designation;
- id;
- instantiated\_function.

#### 4.2.149.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Function\_unit.

The description need not be specified for a particular Function\_unit.

#### 4.2.149.2 extended\_designation

The extended\_designation specifies a structured label for the Function\_unit.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

EXAMPLE 2 IEC 61346-1 specifies designations and structuring principles for Function\_unit objects.

The extended\_designation need not be specified for a particular Function\_unit.

See 4.3.1333 for the application assertion.

#### 4.2.149.3 id



The id specifies the identifier of the Function\_unit.

#### **4.2.149.4 instantiated\_function**

An instantiated\_function specifies the Function\_definition (see 4.2.145) that serves as template for the Function\_unit or Product\_identification (see 4.2.268).

Each instantiated\_function may be one of the following: Function\_definition (see 4.2.145) or Product\_identification (see 4.2.268).

See 4.3.1332 and 4.3.1334 for the application assertions.

#### **4.2.150 Function\_unit\_relationship**

A Function\_unit\_relationship is the relation between two Function\_unit (see 4.2.148) objects.

NOTE The associated Function\_unit (see 4.2.148) objects do not necessarily belong to the same Function\_definition (see 4.2.145) object.

The data associated with an Function\_unit\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

##### **4.2.150.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Function\_unit\_relationship.

The description need not be specified for a particular Function\_unit\_relationship.

##### **4.2.150.2 related**

The related specifies the second of the two Function\_unit (see 4.2.148) objects related by the Function\_unit\_relationship.

See 4.3.1335 for the application assertion.

##### **4.2.150.3 relating**

The relating specifies the first of the two Function\_unit (see 4.2.148) objects related by the Function\_unit\_relationship.

See 4.3.1336 for the application assertion.

##### **4.2.150.4 relation\_type**

## ISO 10303-212:2001(E)

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- derivation;
- implementation;
- redundancy;
- specialization;
- substitution.

NOTE See 4.2.150.4.1 - 4.2.150.4.5 for the definition of each predefined value for `relation_type`.

### 4.2.150.4.1 derivation

derivation: The `Function_unit_relationship` defines a deriving relationship where the related `Function_unit` (see 4.2.148) is based on the relating `Function_unit` (see 4.2.148).

### 4.2.150.4.2 implementation

implementation: The `Function_unit_relationship` defines a relationship where the related `Function_unit` (see 4.2.148) is the realization on the relating `Function_unit` (see 4.2.148).

EXAMPLE 1 A function 'Error handling', specified by a contractor, is implemented by functions 'Monitoring', 'Storage', and 'User interface', taken from a library of standard functions. Each of the standard functions are allocated to Device (see 4.2.88) objects representing the software and hardware components that actually process the error messages.

### 4.2.150.4.3 redundancy

redundancy: The `Function_unit_relationship` defines a relationship where the related `Function_unit` (see 4.2.148) is replicated by the relating `Function_unit` (see 4.2.148).

EXAMPLE 2 The flight controller in an aircraft is built up with a threefold redundancy. The modules compare their output values and can detect possible problems. The modules use different algorithms, thus avoiding that all modules show the same faulty behaviour.

### 4.2.150.4.4 specialization

specialization: The `Function_unit_relationship` defines a relationship where the related `Function_unit` (see 4.2.148) fulfils the requirements of the relating `Function_unit` (see 4.2.148) in a more specific way than defined for the relating `Function_unit` (see 4.2.148).

### 4.2.150.4.5 substitution

substitution: The `Function_unit_relationship` defines a relationship where the related `Function_unit` (see 4.2.148) replaces the relating `Function_unit` (see 4.2.148).

EXAMPLE 3 A Function\_unit (see 4.2.148) is substituted by a Function\_unit (see 4.2.148) of different behaviour.

### 4.2.151 Function\_version

A Function\_version is a version of an Functionality (see 4.2.155) and serves as the collector of the data characterizing a Functionality (see 4.2.155) object in various application contexts.

NOTE 1 An Function\_version may be produced, consumed, used to produce other Function\_version objects, or offered to the market.

NOTE 2 The collection of defining information may be incomplete, i.e., not all of the Function\_definition (see 4.2.145) objects needed to define an Function\_version are associated with the Function\_version.

NOTE 3 The set of Function\_version objects of an Functionality (see 4.2.155) represents the history of the Functionality (see 4.2.155) within a particular life cycle stage or over its complete life cycle.

NOTE 4 An Function\_version may not be referenced by a Function\_definition (see 4.2.145).

The data associated with a Function\_version are the following:

- base\_function;
- description;
- version\_id.

#### 4.2.151.1 base\_function

The base\_function specifies the Functionality (see 4.2.155) with which the Function\_version is associated.

See 4.3.1337 for the application assertion.

#### 4.2.151.2 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Function\_version.

The description need not be specified for a particular Function\_version.

#### 4.2.151.3 version\_id

The version\_id specifies versioning information for the Function\_version.

### 4.2.152 Function\_version\_relationship

A Function\_version\_relationship is the relation between two Function\_version (see 4.2.150) objects.

## ISO 10303-212:2001(E)

The data associated with an `Function_version_relationship` are the following:

- `description`;
- `related`;
- `relating`;
- `relation_type`.

### 4.2.152.1 `description`

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Function_version_relationship`.

The `description` need not be specified for a particular `Function_version_relationship`.

### 4.2.152.2 `related`

The `related` specifies the second of the two `Function_version` (see 4.2.150) objects related by the `Function_version_relationship`.

See 4.3.1338 for the application assertion.

### 4.2.152.3 `relating`

The `relating` specifies the first of the two `Function_version` (see 4.2.150) objects related by the `Function_version_relationship`.

See 4.3.1339 for the application assertion.

### 4.2.152.4 `relation_type`

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- derivation;
- hierarchy;
- sequence;
- supplied function.

NOTE See 4.2.152.4.1 - 4.2.152.4.4 for the definition of each predefined value for `relation_type`.

#### 4.2.152.4.1 derivation

derivation: The `Function_version_relationship` defines a deriving relationship where the related `Function_version` (see 4.2.150) is based on the relating `Function_version` (see 4.2.150) which is an earlier version of the same or of a different `Functionality` (see 4.2.155).

#### 4.2.152.4.1 hierarchy

hierarchy: The `Function_version_relationship` defines a hierarchical relationship where the related `Function_version` (see 4.2.150) is a subversion of the relating `Function_version` (see 4.2.150).

EXAMPLE Revision 1.1 and 1.2 of a document.

#### 4.2.152.4.2 sequence

sequence: The `Function_version_relationship` defines a succession of versions where the relating `Function_version` (see 4.2.150) is the preceding version and the related `Function_version` (see 4.2.150) is the following version. For a `Function_version` (see 4.2.150) there shall be, at the most, one `Function_version_relationship` of this relation type as relating and, at most, one `Function_version_relationship` of this relation type as related.

#### 4.2.152.4.3 supplied function

supplied function: The `Function_version_relationship` defines a relationship between two `Function_version` (see 4.2.150) objects representing the same module in different organizational contexts.

NOTE The different organizational contexts can be represented by different organizational data.

### 4.2.153 Functional\_connectivity\_definition

A `Functional_connectivity_definition` is a specification of the ability to enable the flow of information within a functional module.

Each `Functional_connectivity_definition` is either an `Interface_port_connectivity` (see 4.2.172) or a `Network` (see 4.2.206).

## ISO 10303-212:2001(E)

The data associated with a Functional\_connectivity\_definition are the following:

- connectivity\_of;
- description;
- id;
- version\_id.

### 4.2.153.1 connectivity\_of

The connectivity\_of specifies the Function\_definition (see 4.2.145) object, the internal connectivity of which is specified by Functional\_connectivity\_definition.

See 4.3.1340 for the application assertion.

### 4.2.153.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Functional\_connectivity\_definition.

The description need not be specified for a particular Functional\_connectivity\_definition.

### 4.2.153.3 id

The id specifies the identifier of the Functional\_connectivity\_definition.

### 4.2.153.4 version\_id

The version\_id specifies versioning information for the Functional\_connectivity\_definition.

The version\_id need not be specified for a particular Functional\_connectivity\_definition.

## 4.2.154 Functional\_connectivity\_definition\_relationship

A Functional\_connectivity\_definition\_relationship is the relation between two Functional\_connectivity\_definition (see 4.2.152) objects.

The data associated with an `Functional_connectivity_definition_relationship` are the following:

- `description`;
- `related`;
- `relating`;
- `relation_type`.

#### **4.2.154.1 description**

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Functional_connectivity_definition_relationship`.

The `description` need not be specified for a particular `Functional_connectivity_definition_relationship`.

#### **4.2.154.2 related**

The `related` specifies the second of the two `Functional_connectivity_definition` (see 4.2.152) objects related by the `Functional_connectivity_definition_relationship`.

See 4.3.1341 for the application assertion.

#### **4.2.154.3 relating**

The `relating` specifies the first of the two `Functional_connectivity_definition` (see 4.2.152) objects related by the `Functional_connectivity_definition_relationship`.

See 4.3.1342 for the application assertion.

#### **4.2.154.4 relation\_type**

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution;
- redundancy;
- version hierarchy;
- version sequence.

NOTE 1 See 4.2.154.4.1 - 4.2.154.4.7 for the definition of each predefined value for `relation_type`.

### 4.2.154.4.1 alternate

alternate: The `Functional_connectivity_definition_relationship` defines a relationship where the related `Functional_connectivity_definition` (see 4.2.152) is a possible substitute to the relating `Functional_connectivity_definition` (see 4.2.152).

NOTE 2 This concept refers to the possibility to replace the related `Functional_connectivity_definition` (see 4.2.152). The actual replacement is addressed by 'substitution'.

### 4.2.154.4.2 decomposition

decomposition: The `Functional_connectivity_definition_relationship` defines a relationship where the related `Functional_connectivity_definition` (see 4.2.152) is one of the components into which the relating `Functional_connectivity_definition` (see 4.2.152) is divided.

### 4.2.154.4.3 derivation

derivation: The `Functional_connectivity_definition_relationship` defines a deriving relationship where the related `Functional_connectivity_definition` (see 4.2.152) is based on the relating `Functional_connectivity_definition` (see 4.2.152).

### 4.2.154.4.4 substitution

substitution: The `Functional_connectivity_definition_relationship` defines a relationship where the related `Functional_connectivity_definition` (see 4.2.152) replaces the relating `Functional_connectivity_definition` (see 4.2.152).

### 4.2.154.4.5 redundancy

redundancy: The `Functional_connectivity_definition_relationship` defines a relationship where the related `Functional_connectivity_definition` (see 4.2.152) is replicated by the relating `Functional_connectivity_definition` (see 4.2.152).



EXAMPLE In an aircraft the connectivity from the flight controller to the steering elements is replicated for safety reasons.

#### 4.2.154.4.6 version hierarchy

version hierarchy: The `Functional_connectivity_definition_relationship` defines a hierarchical relationship where the related `Functional_connectivity_definition` (see 4.2.152) is a subversion of the relating `Functional_connectivity_definition` (see 4.2.152).

EXAMPLE Revisions 1.1 and 1.2 of a `Functional_connectivity_definition` (see 4.2.152).

#### 4.2.154.4.7 version sequence

version sequence: The `Functional_connectivity_definition_relationship` defines a succession of versions where the relating `Functional_connectivity_definition` (see 4.2.152) is the preceding version, and the related `Functional_connectivity_definition` (see 4.2.152) is the following version.

### 4.2.155 Functional\_unit\_allocation

A `Functional_unit_allocation` is the relation that specifies the equipment selected to implement the specified `Function_unit` (see 4.2.148).

The data associated with a `Functional_unit_allocation` are the following:

- `allocated_functional_unit`;
- `description`;
- `function_implementation`.

#### 4.2.155.1 allocated\_functional\_unit

The `allocated_functional_unit` specifies the `Function_unit` (see 4.2.148) object that is implemented.

See 4.3.1344 for the application assertion.

#### 4.2.155.2 description

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Functional_unit_allocation`.

The `description` need not be specified for a particular `Functional_unit_allocation`.

#### 4.2.155.3 function\_implementation

The `function_implementation` specifies the equipment that implements a `Function_unit` (see 4.2.148).

Each `function_implementation` may be one of the following: `Device` (see 4.2.88), `Physical_instance` (see 4.2.243), or `Product_component` (see 4.2.265).

See 4.3.1343, 4.3.1345, and 4.3.1346 for the application assertions.

## 4.2.156 Functionality

A Functionality is an action or behaviour by which an electrical product fulfils its purpose.

NOTE A Functionality may be either primitive, i.e., its internal structure is not further described or it may be assembled from other functions. Functionality can be used as a specification or requirement for the implementation or as a functional description of an existing circuit.

The data associated with a Functionality are the following:

- description;
- id;
- name.

### 4.2.156.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Functionality.

The description need not be specified for a particular Functionality.

### 4.2.156.2 id

The id specifies the identifier of the Functionality.

### 4.2.156.3 name

The name specifies a speaking designation of the Functionality.

The name need not be specified for a particular Functionality.

## 4.2.157 General\_classification

A General\_classification is a classification of an object which characterizes all objects of the same kind; such a classification is independent from the application of the classified object.

EXAMPLE 1 This information can be used as a criterion for selecting a specific type of equipment from a component database.

EXAMPLE 2 IEC 61355 specifies a classification system for documents.

EXAMPLE 3 IEC 3B/245/CDV (IEC 61346-2) specifies a classification system for objects.

EXAMPLE 4 IEC 81714-3 specifies a classification system for connect nodes and networks.

EXAMPLE 5 IEC 60529 specifies a classification system for the degree of protection provided by enclosures.

EXAMPLE 6 IEC 60721 specifies a classification system for environmental conditions.

The data associated with a General\_classification are the following:

- classification\_source;
- description;
- id;
- used\_classification\_system;
- version\_id.

#### **4.2.157.1 classification\_source**

The classification\_source specifies a reference under which the specification of the General\_classification can be found.

The classification\_source need not be specified for a particular General\_classification.

Each classification\_source may be one of the following: Class\_reference (see 4.2.43) or External\_library\_reference (see 4.2.132).

See 4.3.1347 and 4.3.1349 for the application assertions.

#### **4.2.157.2 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the General\_classification.

The description need not be specified for a particular General\_classification.

#### **4.2.157.3 id**

The id specifies the identifier of the General\_classification.

#### **4.2.157.4 used\_classification\_system**

The used\_classification\_system specifies the Classification\_system (see 4.2.48) that contains the definition of the classification.

See 4.3.1348 for the application assertion.

#### **4.2.157.5 version\_id**

The version\_id specifies versioning information for the General\_classification.

The version\_id need not be specified for a particular General\_classification.

### **4.2.158 General\_classification\_hierarchy**

A General\_classification\_hierarchy is the specification of a hierarchical structure for General\_classification (see 4.2.156) objects.

## ISO 10303-212:2001(E)

The data associated with a General\_classification\_hierarchy are the following:

- sub\_class;
- super\_class.

### 4.2.158.1 sub\_class

The sub\_class specifies the lower level General\_classification (see 4.2.156) in a classification-hierarchy that is included in the super class.

See 4.3.1350 for the application assertion.

### 4.2.158.2 super\_class

The super\_class specifies the higher level General\_classification (see 4.2.156) in a classification-hierarchy that includes the sub class.

See 4.3.1351 for the application assertion.

## 4.2.159 General\_location\_relationship

A General\_location\_relationship is a type of Location\_relationship (see 4.2.194) that is the relation between two Location (see 4.2.192) objects.

The data associated with an General\_location\_relationship are the following:

- description;
- relation\_type.

### 4.2.159.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the General\_location\_relationship.

The description need not be specified for a particular General\_location\_relationship.

### 4.2.159.2 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- alternate;
- substitution;
- version hierarchy;
- version sequence.

NOTE 1 See 4.2.159.2.1 - 4.2.159.2.3 for the definition of each predefined value for `relation_type`.

#### 4.2.159.2.1 alternate

alternate: The `General_location_relationship` defines a relationship where the related Location (see 4.2.192) is a possible substitute to the relating Location (see 4.2.192).

NOTE 2 This relationship may be used to indicate that two different Location (see 4.2.192) objects specify the same place regardless of rounding errors or the assignment to different location trees.

NOTE 3 This concept refers to the possibility to replace the Location (see 4.2.192). The actual replacement is addressed by 'substitution'.

#### 4.2.159.2.2 substitution

substitution: The `General_location_relationship` defines a relationship where the related Location (see 4.2.192) replaces the relating Location (see 4.2.192).

#### 4.2.159.2.3 version hierarchy

version hierarchy: The `General_location_relationship` defines a hierarchical relationship where the related Location (see 4.2.192) is a subversion of the relating Location (see 4.2.192).

EXAMPLE Revisions 1.1 and 1.2 of a Location (see 4.2.192).

#### 4.2.159.2.4 version sequence

version sequence: The `General_location_relationship` defines a succession of versions where the relating Location (see 4.2.192) is the preceding version, and the related Location (see 4.2.192) is the following version.

### 4.4.2.160 Generic\_note

A `Generic_note` is human-interpretable information that gives further details on a specific thing of interest. By using notes, explanatory information may be added to the product data.

Each `Generic_note` is either a `Note` (see 4.2.210) or a `Set_of_notes` (see 4.2.305).

## ISO 10303-212:2001(E)

The data associated with a `Generic_note` are the following:

- `description`;
- `id`;
- `version_id`.

### 4.2.160.1 `description`

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Generic_note`.

The `description` need not be specified for a particular `Generic_note`.

### 4.2.160.2 `id`

The `id` specifies the identifier of the `Generic_note`.

### 4.2.160.3 `version_id`

The `version_id` specifies versioning information for the `Generic_note`.

The `version_id` need not be specified for a particular `Generic_note`.

### 4.2.161 `Geometrical_tolerance`

A `Geometrical_tolerance` is a type of `Draughting_callout` (see 4.2.117) that is a combination of geometric characteristic symbols, tolerance values, and datum designations, where applicable, to express the permissible variation from the theoretically exact size, profile, orientation, or location of a feature or datum target.

### 4.2.162 `Geometrical_tolerance_symbol`

A `Geometrical_tolerance_symbol` is a type of `Predefined_symbol` (see 4.2.255) that is used to establish a tolerance zone within which the specified conditions of the tolerance apply.

The predefined `Geometrical_tolerance_symbol` that shall be supported by all implementations of this part of ISO 10303 are dependent on the height ( $h$ ) of the text that the symbol accompanies. The height of the characters for the predefined symbols specified here shall be  $h = 2.5$  mm. The graphics shown in Figure 14 are oriented as they appear in a horizontally placed tolerance frame.

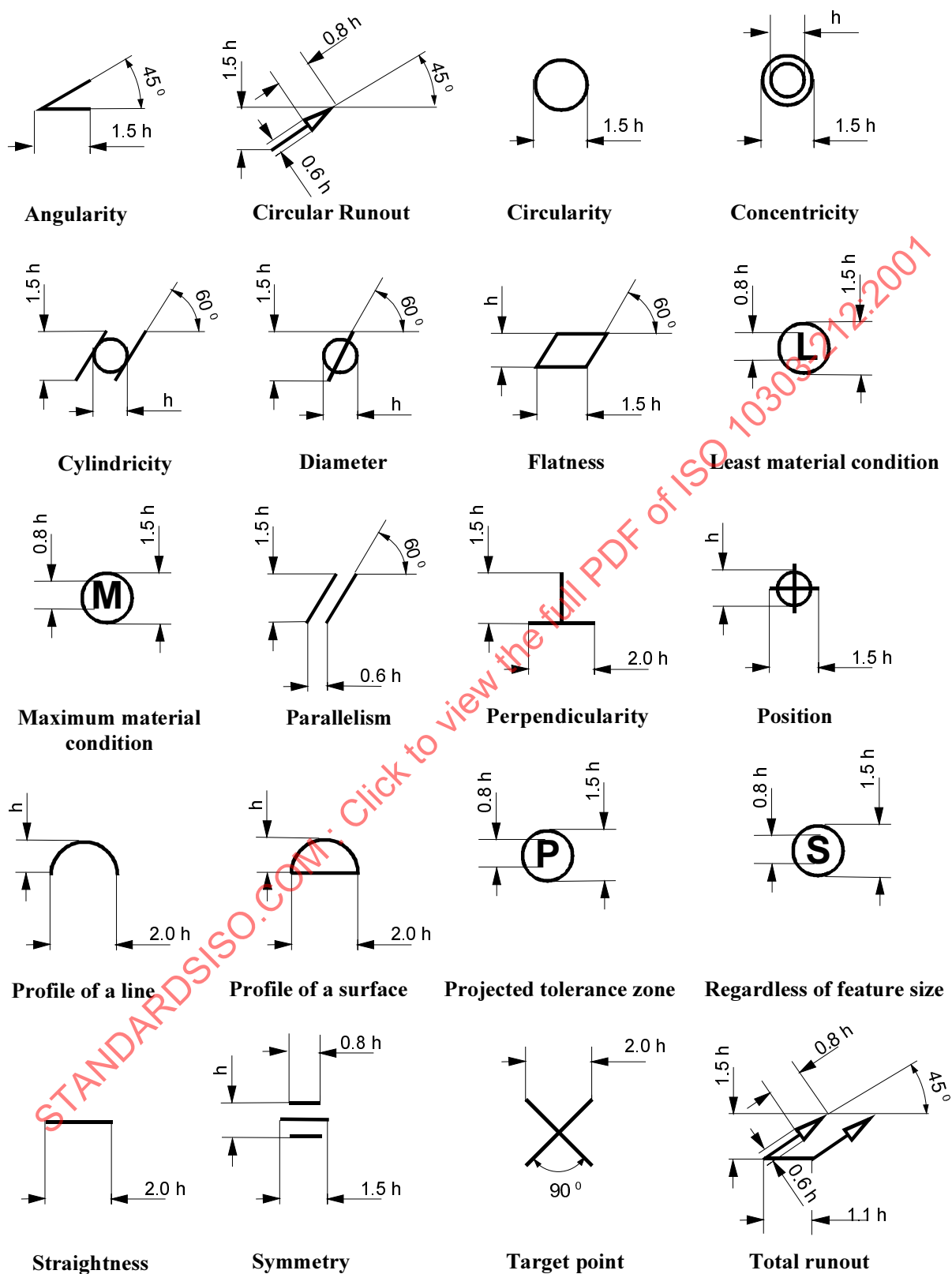


Figure 14 - Predefined geometrical tolerance symbols

## ISO 10303-212:2001(E)

The data associated with a Geometrical\_tolerance\_symbol are the following:

— symbol\_type.

### 4.2.162.1 symbol\_type

The name specifies an alphanumerical string identifying the Geometrical\_tolerance\_symbol in accordance to the definitions given above.

The value of symbol\_type is one of the following:

- angularity;
- circular runout;
- circularity;
- concentricity;
- cylindricity;
- diameter;
- flatness;
- least material condition;
- maximum material condition;
- parallelism;
- perpendicularity;
- position;
- profile of a line;
- profile of a surface;
- projected tolerance zone;
- regardless of feature size;
- straightness;
- symmetry;
- target point;
- total runout.

NOTE See 4.2.162.1.1 - 4.2.162.1.20 for the definition of each permissible value for symbol\_type.



**4.2.162.1.1 angularity**

angularity: An angularity symbol is used to define the condition of a surface or line that is at the specified angle, other than 90 degrees, from a datum plane or axis. An angularity symbol is depicted as two line segments that form an open triangle. The origin of the symbol is the intersection point of the two lines. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.2 circular runout**

circular runout: A circular-runout symbol is used to define the maximum permissible surface variation at any fixed point during one complete rotation of the part about the datum axis. A circular-runout symbol is depicted as a leader and terminated by an arrow. The origin of the symbol is the start of the leader line. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.3 circularity**

circularity: A circularity symbol is used to define the condition of a surface of revolution where all points of the surface intersected by any plane, perpendicular to a common axis or passing through a common centre, are equidistant from the axis. A circularity symbol is depicted as a circle. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.4 concentricity**

concentricity: A concentricity symbol is used to define the condition wherein the axis of all cross sectional elements of a cylinder, cone, or sphere are common to a datum axis. A concentricity symbol is depicted as two concentric circles. The origin of the symbol is the common centre of the circles. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.5 cylindricity**

cylindricity: A cylindricity symbol is used to define two concentric cylinders between which all elements of the specified surface must lie. A cylindricity symbol is depicted as a circle, combined with two tangential and parallel lines. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.6 diameter**

diameter: A diameter symbol is used to indicate that the associated notation applies diametrically. A diameter symbol is depicted as a circle, crossed by a line segment. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.7 flatness**

flatness: A flatness symbol is used to define the allowable perpendicular deviation of surface elements from the plane in which they reside. A flatness symbol is depicted as a parallelogram. The origin of the symbol is the lower, left corner of the parallelogram. The size and graphical representation of the symbol are shown in Figure 14.

#### 4.2.162.1.8 least material condition

least material condition: A least material condition symbol is used to define that the given tolerance applies to the part feature at the tolerance limit where the material content is at its minimum. The symbol is depicted as a circle with the character 'L' positioned in its centre. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

#### 4.2.162.1.9 maximum material condition

maximum material condition: A maximum material condition symbol is used to define that the given tolerance applies to the part feature at the tolerance limit where the material content is at its maximum. A maximum material condition symbol is depicted as a circle with the character 'M' positioned in its centre. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

#### 4.2.162.1.10 parallelism

parallelism: A parallelism symbol is used to define the condition of a surface, axis, or line that is equidistant at all points from a datum plane or axis. A parallelism symbol is depicted as two parallel line segments. The origin of the symbol is the starting point of the first line segment on the left. The size and graphical representation of the symbol are shown in Figure 14.

#### 4.2.162.1.11 perpendicularity

perpendicularity: A perpendicularity symbol is used to define the condition of a surface, axis, or line that is at right angles to a datum plane or axis. A perpendicularity symbol is depicted as two lines, one placed perpendicular to the other. The origin of the symbol is the point of intersection of the two line segments. The size and graphical representation of the symbol are shown in Figure 14.

#### 4.2.162.1.12 position

position: A position symbol is used to define a condition wherein a part or feature has the same contour and is on opposite sides of a central plane, or a condition in which a feature is symmetrically disposed about the central plane of a datum feature. A position symbol is depicted as a circle, crossed by two perpendicular lines, intersected in the centre of the circle. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

#### 4.2.162.1.13 profile of a line

profile of a line: A profile of a line symbol is used to define a tolerance zone, always perpendicular to the profile at points of the profile, within which the specified line must lie. A profile of a line symbol is depicted as an arc. The origin of the symbol is the start point on the left of the arc line. The size and graphical representation of the symbol are shown in Figure 14.

#### 4.2.162.1.14 profile of a surface

profile of a surface: A profile of a surface symbol is used to define a tolerance zone, always perpendicular to the surface, within which the specified surface must lie. A profile of a surface symbol is depicted as a closed arc. The origin of the symbol is the midpoint of the line between the start point and the end point of the arc. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.15 projected tolerance zone**

projected tolerance zone: A projected tolerance zone symbol is used to define the height or depth to which a tolerance of a location applies. A projected tolerance zone symbol is depicted as a circle with the character 'P' positioned at its centre. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.16 regardless of feature size**

regardless of feature size: A regardless of feature size symbol is used to specify that the given tolerance applies to the feature regardless of its size variation. A regardless of feature size symbol is depicted as a circle with the character 'S' positioned in its centre. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.17 straightness**

straightness: A straightness symbol is used to define a tolerance zone within which the considered element must lie and where the element is repressed by a straight line. A straightness symbol is depicted as a straight line segment, horizontal to the tolerance frame. The origin of the symbol is the left end point of the segment. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.18 symmetry**

symmetry: A symmetry symbol is used to define the condition wherein all elements of the feature being toleranced must lie equidistant from the specified datum within the zone defined by the tolerance. A symmetry symbol is depicted as three parallel, horizontal line segments. The origin of the symbol is the midpoint of the middle line segment. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.19 target point**

target point: A target point symbol is used to identify a specific point that is to be used as the datum reference point. A target point is depicted as two perpendicular line segments. The origin of the symbol is the intersection point of the two line segments. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.162.1.20 total runout**

total runout: A total runout symbol is used to define the maximum permissible surface variation of all surface elements during one complete rotation of the part about the datum axis. A total runout symbol is depicted as two parallel leader lines, each terminated by an arrow. The origin of the symbol is the starting point of the leader on the left. The size and graphical representation of the symbol are shown in Figure 14.

**4.2.163 Gis\_position**

A Gis\_position is the positioning and orientation information necessary for transforming coordinate values between a local coordinate space and the global coordinate space of the earth. Transformation procedures depend on the GIS coordinate system.

## ISO 10303-212:2001(E)

EXAMPLE 1 A detailed discussion of global positioning systems is provided in: HOFMANN - WELLENHOF B., LICHTENEGGER H., COLLINS J., *Global Positioning System - Theory and Practice*; Fourth Edition, Springer Verlag Wien, New York, 1997.

The data associated with a `Gis_position` are the following:

- height;
- scale;
- system;
- `x_axis_delta_x`;
- `x_axis_delta_y`;
- `x_coordinate`;
- `y_coordinate`;
- zone.

### 4.2.163.1 height

The height specifies the distance of the origin of the local coordinate system above the zero elevation datum.

NOTE The zero elevation datum, also referred as sea level, is specified by the geodetic authorities for the region in question.

### 4.2.163.2 scale

The scale specifies a transformation factor applied to the conversion of point coordinates between a local coordinate system and a GIS coordinate system. The precise application of the transformation will depend on the GIS system.

### 4.2.163.3 system

The system specifies the identifier of the GIS system being used.

EXAMPLE 2 Gauss-Krueger, Universal Transverse Mercator (UTM), and State Plane are examples of GIS systems used for global positioning.

### 4.2.163.4 `x_axis_delta_x`

The `x_axis_delta_x` specifies the abscissa value of the end point of a vector indicating the positive x axis of GIS coordinate space in the local coordinate system.

### 4.2.163.5 `x_axis_delta_y`

The `x_axis_delta_y` specifies the ordinate value of the end point of a vector indicating the orientation of the positive x axis of the GIS coordinate space in the local coordinate system.

EXAMPLE 3 The GIS coordinate system XY00 has an origin at the intersection of the equator and the Greenwich meridian; the *x\_axis* of the coordinate system runs East (positive) and West (negative); the *y\_axis* runs North (positive) and South (negative); the positive *z\_axis* is up. An *x\_axis\_delta\_x* of 1.0 and *x\_axis\_delta\_y* of 1.0 indicates that the *x\_axis* of the GIS coordinate space makes a  $+45^\circ$  angle with respect to the *x\_axis* of the local coordinate system. If the local coordinate space was superimposed on the GIS coordinate space, the positive *x\_axis* of the local coordinate system would point in a Southeast direction ( $-45^\circ$ ).

#### 4.2.163.6 *x\_coordinate*

The *x\_coordinate* specifies the distance from the *y\_axis* of the coordinate space defined by the GIS system and zone.

#### 4.2.163.7 *y\_coordinate*

The *y\_coordinate* specifies the distance from the *x\_axis* of the coordinate space defined by the GIS system and zone.

#### 4.2.163.8 *zone*

The *zone* specifies a subdivision of the earth's surface based on the GIS system.

EXAMPLE 4 The Gauss-Krueger GIS system subdivides the earth into 120 zones, each of which is  $3^\circ$  in longitudinal width. Each zone is identified as  $3^\circ$ ,  $6^\circ$ ,  $9^\circ$ , etc., from the Greenwich meridian.

### 4.2.164 *Group*

A *Group* is a collection of graphical elements and other previously defined groups generated into related sets.

The data associated with a *Group* are the following:

- description;
- id;
- members.

#### 4.2.164.1 *description*

The *description* specifies an alphanumeric string containing human-interpretable text that gives further details about the *Group*.

The *description* need not be specified for a particular *Group*.

#### 4.2.164.2 *id*

The *id* specifies the identifier of the *Group*.

#### 4.2.164.3 *members*

## ISO 10303-212:2001(E)

The members specifies the annotation elements or subgroups that are contained within the Group.

See 4.3.1352 for the application assertion.

### 4.2.165 Group\_annotation\_element

A Group\_annotation\_element is a type of Group\_element (see 4.2.165) that is the annotation contained in a Group (see 4.2.163).

The data associated with a Group\_annotation\_element are the following:

— basis\_annotation.

#### 4.2.165.1 basis\_annotation

The basis\_annotation specifies the Draughting\_annotation (see 4.2.116) that is contained in the Group (see 4.2.163).

See 4.3.1353 for the application assertion.

### 4.2.166 Group\_element

A Group\_element is an annotation or another Group (see 4.2.163) object that is a member of a Group (see 4.2.163).

Each Group\_element is either a Group\_annotation\_element (see 4.2.164) or a Sub\_group (see 4.2.333).

### 4.2.167 Hardcopy

A Hardcopy is a type of Document\_file (see 4.2.106) that is a document or a portion thereof that is provided in nondigital form.

### 4.2.168 Hatching\_pattern

A Hatching\_pattern is a single, uniformly spaced geometric pattern of lines. The basis of the hatching pattern is an infinite straight line that is repeated across the fill area and clipped to its boundaries.

The data associated with a Hatching\_pattern are the following:

— angle;

— displacement;

— hatch\_line\_appearance.

#### 4.2.168.1 angle

The angle specifies the angular rotation of the curves of the hatching pattern, measured counter-clockwise, from the x-axis of the coordinate system into which the hatching pattern is placed.

### 4.2.168.2 displacement

The displacement specifies a vector that positions the adjacent lines of the hatch pattern from the current line.

### 4.2.168.3 hatch\_line\_appearance

The hatch\_line\_appearance specifies the outlook of the curves used to create a Hatching\_pattern.

See 4.3.1354 for the application assertion.

## 4.2.169 Hierarchical\_location\_relationship

A Hierarchical\_location\_relationship is a type of Location\_relationship (see 4.2.194) that specifies the relationship where the related Location (see 4.2.192) is a sub- location into which the relating Location (see 4.2.192) is divided.

NOTE It is understood that a sublocation is inside the higher-level location.

The data associated with a Hierarchical\_location\_relationship are the following:

- description;
- transformation.

### 4.2.169.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Hierarchical\_location\_relationship.

The description need not be specified for a particular Hierarchical\_location\_relationship.

### 4.2.169.2 transformation

The transformation specifies the geometrical transformation that is used to calculate the exact spatial position of the sublocation within the Location (see 4.2.192).

The transformation need not be specified for a particular Hierarchical\_location\_relationship.

## 4.2.170 Instance\_placement

An Instance\_placement is the information describing how to place a physical or functional item that is defined in its own coordinate space in the coordinate space of a superordinate Product\_component (see 4.2.265).

## ISO 10303-212:2001(E)

The data associated with an Instance\_placement are the following:

- placed\_instance;
- referenced\_product\_component;
- transformation.

### 4.2.170.1 placed\_instance

The placed\_instance specifies the item that is placed.

Each placed\_instance may be one of the following: Single\_device (see 4.2.314) or Single\_function\_unit (see 4.2.315).

See 4.3.1356 and 4.3.1357 for the application assertions.

### 4.2.170.2 referenced\_product\_component

The referenced\_product\_component specifies the superordinate Product\_component (see 4.2.265) that is defined in the reference coordinate space.

See 4.3.1355 for the application assertion.

### 4.2.170.3 transformation

The transformation specifies the geometrical transformation between the local coordinate system of the item that is placed into the reference coordinate system of the associated Product\_component (see 4.2.265).

### 4.2.171 Interface

An Interface specifies Interface\_terminal (see 4.2.174) objects that characterize the use or the intended purpose of a piece of equipment.



NOTE The Interface may provide a possible selection criterion if a given piece of equipment needs to be substituted by a different one. Possible selection criteria may be assigned as Data\_ - element (see 4.2.70) objects to the Interface.

The data associated with an Interface are the following:

- description;
- external\_access;
- id;
- interface\_of.

#### **4.2.171.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Interface.

The description need not be specified for a particular Interface.

#### **4.2.171.2 external\_access**

The external\_access specifies the interface by assigning interface\_terminal (see 4.2.174) objects to the Interface.

See 4.3.1359 for the application assertion.

#### **4.2.171.3 id**

The id specifies the identifier of the Interface.

#### **4.2.171.4 interface\_of**

The interface\_of specifies the Design\_discipline\_item\_definition (see 4.2.86) to which the Interface belongs.

See 4.3.1358 for the application assertion.

#### **4.2.172 Interface\_port**

An Interface\_port defines a single access point for the functionality of a piece of equipment.

The data associated with an Interface\_port are the following:

- description;
- id;
- interface\_port\_of.

#### **4.2.172.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Interface\_port.

The description need not be specified for a particular Interface\_port.

#### **4.2.172.2 id**

The id specifies the identifier of the Interface\_port.

#### **4.2.172.3 interface\_port\_of**

The interface\_port\_of specifies the Function\_definition (see 4.2.145) to which the Interface\_port is assigned.

See 4.3.1360 for the application assertion.

#### **4.2.173 Interface\_port\_connectivity**

An Interface\_port\_connectivity is a type of Functional\_connectivity\_definition (see 4.2.152) that is a link between Interface\_port (see 4.2.171) objects that are connected internally in the functional module.

NOTE The Interface\_port\_connectivity allows to specify internal networks between the access nodes of Function\_definition (see 4.2.145) objects belonging to the same Function\_version (see 4.2.150) without the need to describe the internal structure of the module.

EXAMPLE The internal connectivity of a jumper module can be expressed by using Interface\_port\_connectivity objects.

The data associated with an Interface\_port\_connectivity are the following:

— connected\_interface\_port.

##### **4.2.173.1 connected\_interface\_port**

The connected\_interface\_port specifies the linked Interface\_port (see 4.2.171) objects.

See 4.3.1361 for the application assertion.

#### **4.2.174 Interface\_port\_relationship**

A Interface\_port\_relationship is the relation between two Interface\_port (see 4.2.171) objects.

The data associated with an Interface\_port\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### 4.2.174.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Interface\_port\_relationship.

The description need not be specified for a particular Interface\_port\_relationship.

#### 4.2.174.2 related

The related specifies the second of the two Interface\_port (see 4.2.171) objects related by the Interface\_port\_relationship.

See 4.3.1362 for the application assertion.

#### 4.2.174.3 relating

The relating specifies the first of the two Interface\_port (see 4.2.171) objects related by the Interface\_port\_relationship.

See 4.3.1363 for the application assertion.

#### 4.2.174.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- decomposition;
- derivation.

NOTE See 4.2.174.4.1 - 4.2.174.4.2 for the definition of each predefined value for relation\_type.

#### 4.2.174.4.1 decomposition

decomposition: The Interface\_port\_relationship defines a relationship where the Interface\_port (see 4.2.171) is one of the components into which the relating Interface\_port (see 4.2.171) is decomposed.

#### 4.2.174.4.2 derivation

derivation: The Interface\_port\_relationship defines a deriving relationship where the related Interface\_port (see 4.2.171) is based on the relating Interface\_port (see 4.2.171).

### 4.2.175 Interface\_terminal

An Interface\_terminal defines a single access point for a piece of equipment.

The data associated with an Interface\_terminal are the following:

- description;
- id;
- interface\_terminal\_of;
- maximum\_number\_of\_conductors;
- uses.

#### 4.2.175.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Interface\_terminal.

The description need not be specified for a particular Interface\_terminal.

#### 4.2.175.2 id

The id specifies the identifier of the Interface\_terminal.

#### 4.2.175.3 interface\_terminal\_of

The interface\_terminal\_of specifies the Design\_discipline\_item\_definition (see 4.2.86) to which the Interface\_terminal is assigned.

See 4.3.1364 for the application assertion.

#### 4.2.175.4 maximum\_number\_of\_conductors

The maximum\_number\_of\_conductors specifies the maximum amount of wires or cables that may be connected to the associated Terminal (see 4.2.338).

The maximum\_number\_of\_conductors need not be specified for a particular Interface\_terminal.

### 4.2.175.5 uses

The uses specifies the Terminal (see 4.2.338) in the next lower level of the hierarchy onto which the Interface\_terminal is mapped.

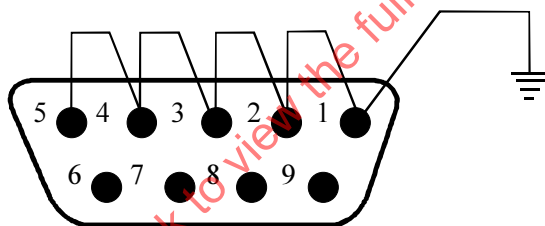
See 4.3.1365 for the application assertion.

### 4.2.176 Interface\_terminal\_connection

An Interface\_terminal\_connection is a type of Connectivity\_definition (see 4.2.61) that is a link between Interface\_terminal (see 4.2.174) objects that are connected internally of the equipment.

**NOTE** The Interface\_terminal\_connection allows one to specify the internal connections between the access nodes of a Design\_discipline\_item\_definition (see 4.2.86) without the need to specify the internal structure of the piece of equipment.

**EXAMPLE** In Figure 15 an internally wired 9-pin connector is shown. Every odd pin is connected to ground. The pins are modelled as Interface\_terminal (see 4.2.174) objects that are linked by four Interface\_terminal\_connection objects. The connection to ground is provided by a Connection (see 4.2.59) object.



**Figure 15 - Internally wired connector**

The data associated with an Interface\_terminal\_connection are the following:

— connected\_interface\_terminal.

#### 4.2.176.1 connected\_interface\_terminal

The connected\_interface\_terminal specifies the linked Interface\_terminal (see 4.2.174) objects.

See 4.3.1366 for the application assertion.

### 4.2.177 Interface\_terminal\_relationship

A Interface\_terminal\_relationship is the relation between two Interface\_terminal (see 4.2.174) objects.

## ISO 10303-212:2001(E)

The data associated with a `Interface_terminal_relationship` are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.177.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the `Interface_terminal_relationship`.

The description need not be specified for a particular `Interface_terminal_relationship`.

### 4.2.177.2 related

The related specifies the second of the two `Interface_terminal` (see 4.2.174) objects related by the `Interface_terminal_relationship`.

See 4.3.1367 for the application assertion.

### 4.2.177.3 relating

The relating specifies the first of the two `Interface_terminal` (see 4.2.174) objects related by the `Interface_terminal_relationship`.

See 4.3.1368 for the application assertion.

### 4.2.177.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- decomposition;
- derivation.

NOTE See 4.2.177.4.1 - 4.2.177.4.2 for the definition of each predefined value for relation\_type.

#### 4.2.177.4.1 decomposition

decomposition: The `Interface_terminal_relationship` defines a relationship where the related `Interface_terminal` (see 4.2.174) is one of the components into which the relating `Interface_terminal` (see 4.2.174) is divided.

**4.2.177.4.2 derivation**

derivation: The `Interface_terminal_relationship` defines a deriving relationship where the related `Interface_terminal` (see 4.2.174) is based on the relating `Interface_terminal` (see 4.2.174).

**4.2.178 Interval\_of\_time**

An `Interval_of_time` is a period of time.

The data associated with a `Interval_of_time` are the following:

- `end_definition`;
- `start_definition`.

**4.2.178.1 end\_definition**

The `end_definition` defines the end of the `Interval_of_time`, either by referring to a bound, or specifying the extend of the `Interval_of_time` by a `Duration` (see 4.2.126). If the `end_definition` refers to an `Event_reference` (see 4.2.130) or `Date_time` (see 4.2.79), this particular bound of the resulting interval is excluded from it.

There shall be exactly one object that defines the `end_definition` for an `Interval_of_time`.

Each `end_definition` may be one of the following: `Date_time` (see 4.2.79), `Duration` (see 4.2.126), or `Event_reference` (see 4.2.130).

See 4.3.1369, 4.3.1371, and 4.3.1372 for the application assertions.

**4.2.178.2 start\_definition**

The `start_definition` defines the beginning of the `Interval_of_time`. The bound specified by the `start_definition` is included in the resulting interval.

There shall be exactly one object that defines the `start_definition` for an `Interval_of_time`.

Each `start_definition` may be one of the following: `Date_time` (see 4.2.79) or `Event_reference` (see 4.2.130).

See 4.3.1370 and 4.3.1373 for the application assertions.

**4.2.179 Item**

An `Item` is a thing produced or intended to be produced, set up, designed, or installed in an electrotechnical system. It can be either a single component or an assembly of arbitrary complexity.

## ISO 10303-212:2001(E)

EXAMPLE The electrotechnical equipment controlling a whole plant can be considered to be an Item.

NOTE Information about Item objects shall not be misused to lay down data that is covered elsewhere in this part of ISO 10303.

The data associated with an Item are the following:

- description;
- extended\_designation;
- id;
- name.

### 4.2.179.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Item.

The description need not be specified for a particular Item.

### 4.2.179.2 extended\_designation

The extended\_designation specifies a label for the Item that may be visualized in schematics.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

The extended\_designation need not be specified for a particular Item.

See 4.3.1374 for the application assertion.

### 4.2.179.3 id

The id specifies the identifier of the Item.

### 4.2.179.4 name

The name specifies a speaking designation of the Item.

## 4.2.180 Item\_definition\_relationship

An Item\_definition\_relationship is a relationship between two Design\_discipline\_item\_definition (see 4.2.86) objects.



The data associated with an Item\_definition\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### 4.2.180.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Item\_definition\_relationship.

The description need not be specified for a particular Item\_definition\_relationship.

#### 4.2.180.2 related

The related specifies the second of the two Design\_discipline\_item\_definition (see 4.2.86) objects related by the Item\_definition\_relationship.

See 4.3.1375 for the application assertion.

#### 4.2.180.3 relating

The relating specifies the first of the two Design\_discipline\_item\_definition (see 4.2.86) objects related by the Item\_definition\_relationship.

See 4.3.1376 for the application assertion.

#### 4.2.180.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- derivation;
- substitution.

NOTE 1 See 4.2.180.4.1 - 4.2.180.4.3 for the definition of each predefined value for relation\_type.

##### 4.2.180.4.1 alternate

alternate: The Item\_definition\_relationship defines a relationship where the related Design\_discipline\_item\_definition (see 4.2.86) is a possible substitute to the relating Design\_discipline\_item\_definition (see 4.2.86).

## ISO 10303-212:2001(E)

NOTE 2 This concept refers to the possibility to replace the Design\_discipline\_item\_definition (see 4.2.86). The actual replacement is addressed by 'substitution'.

### 4.2.180.4.2 derivation

derivation: The Item\_definition\_relationship defines a deriving relationship where the related Design\_discipline\_item\_definition (see 4.2.86) is based on the relating Design\_discipline\_item\_definition (see 4.2.86).

### 4.2.180.4.3 substitution

substitution: The Item\_definition\_relationship defines a relationship where the related Design\_discipline\_item\_definition (see 4.2.86) replaces the relating Design\_discipline\_item\_definition (see 4.2.86).

## 4.2.181 Item\_identification

An Item\_identification is a sequence of alphanumeric characters that labels an Item (see 4.2.178).

NOTE 1 If equipment is to be repaired or exchanged, the Item\_identification allows the determination of the exact type of equipment.

NOTE 2 An Item\_identification of an Item (see 4.2.178) can be unique to each organization.

EXAMPLE The order number may serve as an Item\_identification.

The data associated with an Item\_identification are the following:

- coding\_type;
- id.

### 4.2.181.1 coding\_type

The coding\_type specifies the syntax used for the id.

The coding\_type need not be specified for a particular Item\_identification.

See 4.3.1377 for the application assertion.

### 4.2.181.2 id

The id specifies the identifier of the Item\_identification.

## 4.2.182 Item\_presentation

An Item\_presentation is the association of an item to its presentation.

The data associated with an Item\_presentation are the following:

- id;
- presented\_item;
- presenting\_item.

#### 4.2.182.1 id

The id specifies the identifier of the Item\_presentation.

The id need not be specified for a particular Item\_presentation.

#### 4.2.182.2 presented\_item

The presented\_item specifies the item that is presented.

Each presented\_item may be one of the following: Activity (see 4.2.1), Activity\_element (see 4.2.2), Alias\_identification (see 4.2.9), Classification\_association (see 4.2.46), Classification\_attribute (see 4.2.47), Classification\_system (see 4.2.48), Connecting\_line (see 4.2.58), Connectivity\_definition (see 4.2.61), Cross\_reference (see 4.2.64), Data\_element (see 4.2.70), Data\_element\_definition (see 4.2.72), Data\_element\_specification (see 4.2.75), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Document (see 4.2.101), Document\_version (see 4.2.114), Free\_segment (see 4.2.144), Function\_definition (see 4.2.145), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_version (see 4.2.150), Functional\_connectivity\_definition (see 4.2.152), Functionality (see 4.2.155), General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_identification (see 4.2.180), Item\_version (see 4.2.182), Location (see 4.2.192), Node (see 4.2.208), Notification (see 4.2.215), Object\_designation (see 4.2.217), Page\_connector\_presentation (see 4.2.228), Path (see 4.2.232), Path\_node (see 4.2.233), Physical\_instance (see 4.2.243), Port (see 4.2.247), Process\_variable (see 4.2.260), Project (see 4.2.271), Route (see 4.2.290), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_value (see 4.2.313), Technical\_system (see 4.2.336), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.1378, 4.3.1379, 4.3.1380, 4.3.1382, 4.3.1383, 4.3.1384, 4.3.1385, 4.3.1386, 4.3.1387, 4.3.1388, 4.3.1389, 4.3.1390, 4.3.1391, 4.3.1392, 4.3.1393, 4.3.1394, 4.3.1395, 4.3.1396, 4.3.1397, 4.3.1398, 4.3.1399, 4.3.1400, 4.3.1401, 4.3.1402, 4.3.1403, 4.3.1404, 4.3.1405, 4.3.1406, 4.3.1407, 4.3.1408, 4.3.1409, 4.3.1410, 4.3.1411, 4.3.1412, 4.3.1413, 4.3.1414, 4.3.1415, 4.3.1416, 4.3.1417, 4.3.1418, 4.3.1419, 4.3.1420, 4.3.1421, 4.3.1422, 4.3.1423, 4.3.1424, 4.3.1425, 4.3.1426, 4.3.1427, 4.3.1428, 4.3.1429, 4.3.1430, 4.3.1431, 4.3.1432, and 4.3.1433 for the application assertions.

#### 4.2.182.3 presenting\_item

The presenting\_item specifies the Annotation\_element (see 4.2.15) used to display the item.

See 4.3.1381 for the application assertion.

#### 4.2.183 Item\_version

## ISO 10303-212:2001(E)

An `Item_version` is a particular version of an `Item` (see 4.2.178).

NOTE Several versions for the same `Item` (see 4.2.178) may exist at one point in time. The information about valid and invalid versions is handled by the associated organizational data.

The data associated with an `Item_version` are the following:

- `associated_item`;
- `description`;
- `version_id`.

### 4.2.183.1 `associated_item`

The `associated_item` specifies the associated `Item` (see 4.2.178) object.

See 4.3.1434 for the application assertion.

### 4.2.183.2 `description`

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Item_version`.

The `description` need not be specified for a particular `Item_version`.

### 4.2.183.3 `version_id`

The `version_id` specifies versioning information for the `Item_version`.

## 4.2.184 `Item_version_relationship`

An `Item_version_relationship` is a relationship between two `Item_version` (see 4.2.182) objects.

The data associated with an `Item_version_relationship` are the following:

- `description`;
- `related`;
- `relating`;
- `relation_type`.

### 4.2.184.1 `description`

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Item_version_relationship`.

The `description` need not be specified for a particular `Item_version_relationship`.

### 4.2.184.2 `related`

The related specifies the second of the two Item\_version (see 4.2.182) objects related by the Item\_version\_relationship.

See 4.3.1435 for the application assertion.

### 4.2.184.3 relating

The relating specifies the first of the two Item\_version (see 4.2.182) objects related by the Item\_version\_relationship.

See 4.3.1436 for the application assertion.

### 4.2.184.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- derivation;
- hierarchy;
- sequence;
- supplied item.

NOTE See 4.2.184.4.1 - 4.2.184.4.4 for the definition of each predefined value for relation\_type.

#### 4.2.184.4.1 derivation

derivation: The Item\_version\_relationship defines a deriving relationship where the related Item\_version (see 4.2.182) is based on the relating Item\_version (see 4.2.182) which is an earlier version of the same or of a different Item (see 4.2.178).

#### 4.2.184.4.2 hierarchy

hierarchy: The Item\_version\_relationship defines a hierarchical relationship where the related Item\_version (see 4.2.182) is a subversion of the relating Item\_version (see 4.2.182).

EXAMPLE Revision 1.1 and 1.2 of a drive.

#### 4.2.184.4.3 sequence

sequence: The Item\_version\_relationship defines a succession of versions where the relating Item\_version (see 4.2.182) is the preceding version, and the related Item\_version (see 4.2.182) is the following version. For an Item\_version (see 4.2.182), there shall be, at the most, one Item\_version\_relationship of this relation type as relating and, at most, one Item\_version\_relationship of this relation type as related.

#### 4.2.184.4.4 supplied item

## ISO 10303-212:2001(E)

supplied item: The Item\_version\_relationship defines a relationship between two Item\_version (see 4.2.182) objects representing the same item in different organizational contexts.

EXAMPLE A piece of equipment that is identified in a company by the identifier 'C425' and version\_id 'V2.0' is identified by its supplier as '2X45', version\_id 'V3.2'.

### 4.2.185 Language

The Language specifies a language spoken by human beings to communicate with each other verbally or in written form.

The data associated with a Language are the following:

- country\_code;
- language\_code.

#### 4.2.185.1 country\_code

The country\_code specifies the country, as addition to the language, according to the alpha-2 code specified in ISO 3166-1.

EXAMPLE Possible values for country\_code are, e.g., 'GB' for the United Kingdom or 'US' for the United States of America.

The country\_code need not be specified for a particular Language.

#### 4.2.185.2 language\_code

The language\_code specifies the language of the text information in the Alpha-3 bibliographic code specified in ISO 639-2.

EXAMPLE Possible values for language\_code are, e.g., 'eng' for English, 'fre' for French, 'rus' for Russian, or 'ger' for German.

### 4.2.186 Layer

A Layer is a mechanism for the organization of CAD elements. Within the CAD model, one or more levels are positioned in a stacked arrangement. These levels may have specific, defined uses and may be displayed as desired by the user.

The data associated with a Layer are the following:

- description;
- element\_visibility;
- layer\_id.

#### **4.2.186.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Layer.

The description need not be specified for a particular Layer.

#### **4.2.186.2 element\_visibility**

The element\_visibility specifies whether or not the elements assigned to the layer are displayed in the visual presentation of the drawing.

See 4.3.1437 for the application assertion.

#### **4.2.186.3 layer\_id**

The layer\_id specifies the identification of a particular layer.

### **4.2.187 Leader**

A Leader is a type of Directed\_curve (see 4.2.99) that directs a dimension, a note, or a symbol to the intended place or point on a feature appearing on the drawing.

The data associated with a Leader are the following:

- target\_element.

#### **4.2.187.1 target\_element**

The target\_element specifies the annotation element that is the target for the leader.

The target\_element need not be specified for a particular Leader.

See 4.3.1438 for the application assertion.

### **4.2.188 Leader\_directed\_dimension**

A Leader\_directed\_dimension is a type of Dimension (see 4.2.93) that is the graphical presentation of a dimension value and is guided to the feature being dimensioned with a leader.

## ISO 10303-212:2001(E)

The data associated with a `Leader_directed_dimension` are the following:

— `component`.

### 4.2.188.1 `component`

The `component` specifies the leader that visually directs the callout to the element.

See 4.3.1439 for the application assertion.

### 4.2.189 `Leader_terminator`

A `Leader_terminator` is a graphical symbol that is applied to a leader and used to identify the endpoint or point of application of the directed annotation.

The data associated with a `Leader_terminator` are the following:

— `line`;

— `symbol`.

#### 4.2.189.1 `line`

The `line` specifies the leader to which the symbol applies.

See 4.3.1441 for the application assertion.

#### 4.2.189.2 `symbol`

The `symbol` specifies an `Annotation_symbol` (see 4.2.20) that is used to identify the endpoint or point of application of the directed annotation.

See 4.3.1440 for the application assertion.

### 4.2.190 `Line_font`

A `Line_font` is a defined pattern of visible and invisible segments applied to a curve in a repetitive manner.

Each `Line_font` is either an `Externally_defined_line_font` (see 4.2.134), a `Predefined_line_font` (see 4.2.254), or a `User_defined_line_font` (see 4.2.353).

### 4.2.191 `Linear_dimension`

A `Linear_dimension` is a type of `Dimension` (see 4.2.93) that is the graphical presentation of a value of linear distance measured between two points along a straight path.



The data associated with a Linear\_dimension are the following:

- component;
- extent.

#### **4.2.191.1 component**

The component specifies the projection lines that shows the extension of the points, lines, or surfaces that bound the measurement.

See 4.3.1443 for the application assertion.

#### **4.2.191.2 extent**

The extent specifies the dimension line that graphically presents where the dimension value applies.

See 4.3.1442 for the application assertion.

#### **4.2.192 Linear\_pattern\_location**

A Linear\_pattern\_location is a means to locate repeating patterns along the centre-line of a product.

The data associated with a Linear\_pattern\_location are the following:

- distance;
- orientation;
- start\_location.

##### **4.2.192.1 distance**

The distance specifies the span between two neighbouring patterns.

##### **4.2.192.2 orientation**

The orientation specifies the direction of the positive z axis of the Linear\_pattern\_location.

The orientation need not be specified for a particular Linear\_pattern\_location.

##### **4.2.192.3 start\_location**

The start\_location specifies the location of the first pattern.

See 4.3.1444 for the application assertion.

#### **4.2.193 Location**

A Location is a region of space.

## ISO 10303-212:2001(E)

NOTE 1 A Location may provide specific environmental conditions that applies to the equipment that is situated at the Location.

NOTE 2 A Location can be used to reserve space for equipment that is not known yet. Therefore it can be appropriate to define a Location without assigning physical or abstract items to it.

NOTE 3 A Location can be used to support mechanical interference detection.

The data associated with a Location are the following:

- defining\_item;
- description;
- extended\_designation;
- id;
- local\_coordinate\_space;
- position;
- version\_id.

### 4.2.193.1 defining\_item

The defining\_item specifies that the Location has the shape and size of the object which is associated through the 'defining\_item' attribute.

NOTE 4 The shape definition of an equipment item used to specify the shape of the Location needs to be unambiguous.

NOTE 5 The shape need not be explicitly specified through an associated Shape (see 4.2.306) object.

Each defining\_item may be one of the following: Design\_discipline\_item\_definition (see 4.2.86), Physical\_instance (see 4.2.243), Product\_component (see 4.2.265), Single\_device (see 4.2.314), Specified\_device (see 4.2.328), or Technical\_system (see 4.2.336).

See 4.3.1447, 4.3.1450, 4.3.1451, 4.3.1452, 4.3.1453, and 4.3.1454 for the application assertions.

### 4.2.193.2 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Location.

The description need not be specified for a particular Location.

### 4.2.193.3 extended\_designation

The extended\_designation specifies a structured label for the Location.

NOTE 6 The label assigned through `extended_designation` shall be identical to the label assigned by the 'id' attribute.

The `extended_designation` need not be specified for a particular Location.

See 4.3.1449 for the application assertion.

#### **4.2.193.4 id**

The `id` specifies the identifier of the Location.

#### **4.2.193.5 local\_coordinate\_space**

The `local_coordinate_space` specifies the coordinate system of the Location.

The `local_coordinate_space` need not be specified for a particular Location.

See 4.3.1445 for the application assertion.

#### **4.2.193.6 position**

The `position` specifies the spatial position of the Location. If a local coordinate system exists, the 'position' attributes specifies its origin.

The `position` need not be specified for a particular Location.

Each `position` may be one of the following: `Cartesian_point` (see 4.2.37) or `Gis_position` (see 4.2.162).

See 4.3.1446 and 4.3.1448 for the application assertions.

#### **4.2.193.7 version\_id**

The `version_id` specifies versioning information for the Location.

The `version_id` need not be specified for a particular Location.

#### **4.2.194 Location\_assignment**

A `Location_assignment` is an association of a Location (see 4.2.192) with an physical or abstract item.

## ISO 10303-212:2001(E)

The data associated with a Location\_assignment are the following:

- assigned\_location;
- associated\_item;
- description;
- role.

### 4.2.194.1 assigned\_location

The assigned\_location specifies the associated Location (see 4.2.192) object.

See 4.3.1461 for the application assertion.

### 4.2.194.2 associated\_item

The associated\_item specifies the item that is related to the Location (see 4.2.192).

Each associated\_item may be one of the following: Connectivity\_definition (see 4.2.61), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Function\_definition (see 4.2.145), Function\_unit (see 4.2.148), Functional\_connectivity\_definition (see 4.2.152), Physical\_instance (see 4.2.243), Port (see 4.2.247), Product\_component (see 4.2.265), Signal (see 4.2.309), Technical\_system (see 4.2.336), or Terminal (see 4.2.338).

See 4.3.1455, 4.3.1456, 4.3.1457, 4.3.1458, 4.3.1459, 4.3.1460, 4.3.1462, 4.3.1463, 4.3.1464, 4.3.1465, 4.3.1466, and 4.3.1467 for the application assertions.

### 4.2.194.3 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Location\_assignment.

The description need not be specified for a particular Location\_assignment.

### 4.2.194.4 role

The role specifies the relationship between the location and the accommodated item. The value is either user defined or predefined.

The predefined value of role is one of the following:

- accommodation.

NOTE See 4.2.194.4.1 for the definition of each predefined value for role.

#### 4.2.194.4.1 accommodation

accommodation: The accommodation specifies that the referenced item is situated in the associated location.

## 4.2.195 Location\_relationship

A Location\_relationship is the relation between two Location (see 4.2.192) objects.

Each Location\_relationship is either a General\_location\_relationship (see 4.2.158), a Hierarchical\_location\_relationship (see 4.2.168), or a Neighbourhood\_location\_relationship (see 4.2.205).

The data associated with an Location\_relationship are the following:

— related;

— relating.

### 4.2.195.1 related

The related specifies the second of the two Location (see 4.2.192) objects related by the Location\_relationship.

See 4.3.1468 for the application assertion.

### 4.2.195.2 relating

The relating specifies the first of the two Location (see 4.2.192) objects related by the Location\_relationship.

See 4.3.1469 for the application assertion.

## 4.2.196 Logical\_value

A Logical\_value is the representation of a 'true', 'false', or 'unknown' value.

The data associated with a Logical\_value are the following:

— value\_of\_logical\_value.

### 4.2.196.1 value\_of\_logical\_value

The value\_of\_logical\_value specifies the value of a logical variable.

## 4.2.197 Lot\_configuration

A Lot\_configuration is a type of Manufacturing\_configuration (see 4.2.198) that is planned to apply from a given batch of aspects of a part.

The data associated with a Lot\_configuration are the following:

— lot\_id;

— lot\_size.

### 4.2.197.1 lot\_id

## ISO 10303-212:2001(E)

The `lot_id` specifies the identification of the batch of aspects of a part to which the `Lot_configuration` applies.

### 4.2.197.2 `lot_size`

The `lot_id` specifies the size of the batch of aspects of a part that the `Lot_configuration` applies to.

### 4.2.198 `Make_from_relationship`

A `Make_from_relationship` is a type of `Item_definition_relationship` (see 4.2.179) that is a relationship between a `Design_discipline_item_definition` (see 4.2.86) which provides the definition of a raw material, or of a semi-finished part and a `Design_discipline_item_definition` (see 4.2.86) which provides the definition of an object manufactured out of that material, or semi-finished part. The inherited attribute 'related' specifies the raw material or the semi-finished part, and the inherited attribute 'relating' specifies the manufactured part.

### 4.2.199 `Manufacturing_configuration`

A `Manufacturing_configuration` is the association of a `Product_identification` (see 4.2.268), `Product_design` (see 4.2.267), or an `Item_version` (see 4.2.182) with a `Device` (see 4.2.88).

Each `Manufacturing_configuration` is either a `Dated_configuration` (see 4.2.82), a `Lot_configuration` (see 4.2.196), or a `Serial_configuration` (see 4.2.304).

The data associated with a `Manufacturing_configuration` are the following:

- `concerned_organization`;
- `configured_element`;
- `is_solution_for`.

#### 4.2.199.1 `concerned_organization`

The `concerned_organization` specifies the `Organization` (see 4.2.223) in which the `Manufacturing_configuration` is valid. The case where the `concerned_organization` is an empty set means that the `Manufacturing_configuration` regards any organization that may consider the 'configured\_element'.

See 4.3.1471 for the application assertion.

#### 4.2.199.2 `configured_element`

The `configured_element` specifies an `Device` (see 4.2.88) that is controlled by a `Manufacturing_configuration`.

See 4.3.1470 for the application assertion.

#### 4.2.199.3 `is_solution_for`

The `is_solution_for` specifies the characteristic or combination of characteristics for which the `Device` (see 4.2.88) provides a technical solution.

See 4.3.1472 for the application assertion.

## 4.2.200 Marking

A Marking is a sign applied to a piece of equipment used to give additional information about the part.

The data associated with a Marking are the following:

- character\_string;
- figure;
- marked\_device;
- marking\_system;
- meaning;
- position.

### 4.2.200.1 character\_string

The character\_string specifies the list of characters that compose the marking applied to the piece of equipment.

NOTE If the marking consists of pictorial elements the 'character\_string' attribute contains text that identifies the pictures.

EXAMPLE 1 A wire is marked with W001 which is the designator of this particular wire. The 'character\_string' attribute contains the string 'W001'.

EXAMPLE 2 A high voltage cable is marked with a warning symbol. The 'character\_string' attribute contains the string 'Warning label 47-BA'.

### 4.2.200.2 figure

The figure specifies the appearance of the Marking, which includes information about the colour and form of the Marking.

The figure need not be specified for a particular Marking.

See 4.3.1473 for the application assertion.

### 4.2.200.3 marked\_device

The marked\_device specifies the Device (see 4.2.88) the Marking is applied to.

See 4.3.1479 for the application assertion.

#### **4.2.200.4 marking\_system**

The marking\_system specifies the Classification\_system (see 4.2.48) that shall be used to interpret the Marking.

See 4.3.1475 for the application assertion.

#### **4.2.200.5 meaning**

The meaning specifies the content of the Marking.

Each meaning may be one of the following: Data\_element (see 4.2.70) or Generic\_note (see 4.2.159).

See 4.3.1476 and 4.3.1477 for the application assertions.

#### **4.2.200.6 position**

The position specifies the spot where the Marking is applied to the Device (see 4.2.88).

Each position may be one of the following: Cartesian\_point (see 4.2.37) or Linear\_pattern\_location (see 4.2.191).

See 4.3.1474 and 4.3.1478 for the application assertions.

### **4.2.201 Mass**

A Mass is the quantity of matter a component contains.

The data associated with a Mass are the following:

— value\_of\_mass.

#### **4.2.201.1 value\_of\_mass**

The value\_of\_mass specifies the value that indicates the mass.

See 4.3.1480 for the application assertion.

### **4.2.202 Material**

A Material is the matter from which a component is made.

The data associated with a Material are the following:

— substance.

#### **4.2.202.1 substance**

The substance specifies human-interpretable information characterizing the matter.

### **4.2.203 Model\_placed\_annotation**



A `Model_placed_annotation` is a type of `Draughting_annotation` (see 4.2.116) that is located in the coordinate system of the draughting shape model and is subject to view transformations for display.

The data associated with a `Model_placed_annotation` are the following:

- `annotation_layers`;
- `annotation_visibility`.

#### 4.2.203.1 `annotation_layers`

The `annotation_layers` specifies the layers that contain the annotation.

See 4.3.1481 for the application assertion.

#### 4.2.203.2 `annotation_visibility`

The `annotation_visibility` specifies whether or not each piece of annotation placed within the draughting shape model is visible.

See 4.3.1482 for the application assertion.

#### 4.2.204 `Mounting_features`

A `Mounting_features` is the description of the methods that can be used to attach one component to its counterpart.

The data associated with a `Mounting_features` are the following:

- `fixture`.

#### 4.2.204.1 `fixture`

The `fixture` specifies human-interpretable information describing the mounting features.

#### 4.2.205 `Multi_language_note`

A `Multi_language_note` is a type of `Set_of_notes` (see 4.2.305) that is a group of `Note` (see 4.2.210) objects. Each `Note` (see 4.2.210) is authored in its individual language. In all cases the meaning of the text contained in the `'text_of_note'` attribute of the grouped `Note` (see 4.2.210) objects is the same.

**NOTE** A `Multi_language_note` shall only be associated to an attribute, not to a whole item.

**EXAMPLE** A `Multi_language_note` grouping two `Note` (see 4.2.210) objects with the content `'Attention - High voltage'` and `'Achtung - Hochspannung'`.

#### 4.2.206 `Neighbourhood_location_relationship`

A `Neighbourhood_location_relationship` is a type of `Location_relationship` (see 4.2.194) that is the relation between two `Location` (see 4.2.192) objects that specifies the spatial arrangement between the related `Location` (see 4.2.192) objects.

## ISO 10303-212:2001(E)

The data associated with an `Neighbourhood_location_relationship` are the following:

— `relation_type`.

### 4.2.206.1 `relation_type`

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

NOTE The relating Location (see 4.2.192) is a part of the related Location (see 4.2.192).

The predefined value of `relation_type` is one of the following:

- `above`;
- `across`;
- `adjoining`;
- `around`;
- `outside`.

NOTE See 4.2.206.1.1 - 4.2.206.1.5 for the definition of each predefined value for `relation_type`.

#### 4.2.206.1.1 `above`

`above`: The `Neighbourhood_location_relationship` defines a relationship where the related Location (see 4.2.192) is on top of the relating Location (see 4.2.192).

#### 4.2.206.1.2 `across`

`across`: The `Neighbourhood_location_relationship` defines a relationship where the related Location (see 4.2.192) goes from one side to another side of the relating Location (see 4.2.192).

EXAMPLE A high-voltage cable may run across a private property.

#### 4.2.206.1.3 `adjoining`

`adjoining`: The `Neighbourhood_location_relationship` defines a relationship where the related Location (see 4.2.192) is located at the boundary of the relating Location (see 4.2.192).

#### 4.2.206.1.4 `around`

`around`: The `Neighbourhood_location_relationship` defines a relationship where the related Location (see 4.2.192) specifies an area that surrounds the relating Location (see 4.2.192).

NOTE The relating Location (see 4.2.192) is not a part of the related Location (see 4.2.192).

#### 4.2.206.1.5 outside

outside: The `Neighbourhood_location_relationship` defines a deriving relationship where the related Location (see 4.2.192) is external the relating Location (see 4.2.192).

### 4.2.207 Network

A Network is a type of `Functional_connectivity_definition` (see 4.2.152) that is the link between Port (see 4.2.247) objects with the intention to allow the flow of information.

NOTE A Network may be decomposed into its constituents. The information about the constituents is specified through `Functional_connectivity_definition_relationship` (see 4.2.153) of type 'decomposition'.

EXAMPLE 2 bit wide parallel bus is described as a Network consisting of 32 underlying Network objects. The lower level Network objects are related to the upper level Network object through `Functional_connectivity_definition_relationship` (see 4.2.153) objects of type 'decomposition'.

The data associated with a Network are the following:

— `connected_port`.

#### 4.2.207.1 connected\_port

The `connected_port` specifies the Port (see 4.2.247) objects that are joined with each other within a Network.

See 4.3.1483 for the application assertion.

### 4.2.208 Next\_higher\_assembly

A `Next_higher_assembly` is a type of `Assembly_component_relationship` (see 4.2.26) that is a relation where the attribute `related` specifies a constituent of an assembly, and the attribute `relating` specifies the immediate parent assembly of the constituent. A constituent may be a single part or an assembly.

EXAMPLE The assembly 'motor' may be assembled of single parts. The belt conveyor assembly is formed of assemblies.

### 4.2.209 Node

A Node is a named position that is of interest for the placement of equipment.

NOTE 1 A Route (see 4.2.290) is understood as a sequence of vertices and edges, where the vertices are Node objects.

NOTE 2 A Node does not have real world coordinates unless a `Path_node` (see 4.2.233) assigns them to the Node.

## ISO 10303-212:2001(E)

The data associated with a Node are the following:

- assigned\_location;
- description;
- id;
- implemented\_by;
- position.

### 4.2.209.1 assigned\_location

The assigned\_location specifies the Location (see 4.2.192) that contains the Node.

See 4.3.1485 for the application assertion.

### 4.2.209.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Node.

The description need not be specified for a particular Node.

### 4.2.209.3 id

The id specifies the identifier of the Node.

### 4.2.209.4 implemented\_by

The implemented\_by specifies the Device (see 4.2.88) objects that are used to implement the Node.

EXAMPLE A node can be implemented by elements such as cable clips or fasteners.

See 4.3.1484 for the application assertion.

### 4.2.209.5 position

The position specifies the placement of the Node.

The position need not be specified for a particular Node.

See 4.3.1486 for the application assertion.

### 4.2.210 Node\_relationship

A Node\_relationship is the relation between two Node (see 4.2.208) objects.

The data associated with an Node\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### **4.2.210.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Node\_relationship.

The description need not be specified for a particular Node\_relationship.

#### **4.2.210.2 related**

The related specifies the second of the two Node (see 4.2.208) objects related by the Node\_relationship.

See 4.3.1487 for the application assertion.

#### **4.2.210.3 relating**

The relating specifies the first of the two Node (see 4.2.208) objects related by the Node\_relationship.

See 4.3.1488 for the application assertion.

#### **4.2.210.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- `alternate`;
- `decomposition`;
- `derivation`;
- `substitution`.

NOTE 1 See 4.2.210.4.1 - 4.2.210.4.4 for the definition of each predefined value for `relation_type`.

### 4.2.210.4.1 `alternate`

`alternate`: The `Node_relationship` defines a relationship where the related Node (see 4.2.208) is a possible substitute to the relating Node (see 4.2.208).

NOTE 2 This allows one to map Node (see 4.2.208) objects onto each other.

NOTE 3 This concept refers to the possibility to replace the Node (see 4.2.208). The actual replacement is addressed by 'substitution'.

### 4.2.210.4.2 `decomposition`

`decomposition`: The `Node_relationship` defines a relationship where the related Node (see 4.2.208) is one of the components into which the relating Node (see 4.2.208) is divided up.

### 4.2.210.4.3 `derivation`

`derivation`: The `Node_relationship` defines a deriving relationship where the related Node (see 4.2.208) is based on the relating Node (see 4.2.208).

### 4.2.210.4.4 `substitution`

`substitution`: The `Node_relationship` defines a relationship where the related Node (see 4.2.208) replaces the relating Node (see 4.2.208).

## 4.2.211 Note

A Note is a type of `Generic_note` (see 4.2.159) that is human-interpretable information intended to give further details on a specific part of an electrotechnical system. By using notes, explanatory information can be added to the product data.

The data associated with a Note are the following:

- `kind`;
- `language_specification`;
- `text_of_note`.

### 4.2.211.1 `kind`

The kind specifies the category of the Note. The value is either user defined or predefined.

The predefined value of kind is one of the following:

- additional language dependent string;
- assembly instruction;
- explanatory note;
- installation instruction;
- manufacturing instruction;
- operating instruction;
- primary language dependent string.

NOTE See 4.2.211.1.1 - 4.2.211.1.7 for the definition of each predefined value for kind.

#### **4.2.211.1.1 additional language dependent string**

additional language dependent string: The textual information of the Note is phrased in a language other than the primary language.

#### **4.2.211.1.2 assembly instruction**

assembly instruction: The Note contains information about the assembly process of the equipment to which the Note is assigned.

#### **4.2.211.1.3 explanatory note**

explanatory note: The Note contains additional descriptive information about the data.

#### **4.2.211.1.4 installation instruction**

installation instruction: The Note contains information about the erection of the equipment.

#### **4.2.211.1.5 manufacturing instruction**

manufacturing instruction: The Note contains information about the manufacturing of the equipment.

#### **4.2.211.1.6 operating instruction**

operating instruction: The Note contains information about the functioning or handling of the equipment.

#### **4.2.211.1.7 primary language dependent string**

primary language dependent string: The textual information of the Note is phrased in the original language.

#### **4.2.211.2 language\_specification**

## ISO 10303-212:2001(E)

The language\_specification specifies a language spoken by human beings to communicate with each other verbally or in written form. The language symbol given in ISO 639 shall be used.

See 4.3.1489 for the application assertion.

### 4.2.211.3 text\_of\_note

The text\_of\_note specifies human-interpretable information that shall be conveyed to the user.

### 4.2.212 Note\_association

The Note\_association is the relation between the annotation information and the physical or abstract item that is described. The purpose of this relationship is specified by the content of the attributes applied\_attribute\_name and role.

The data associated with a Note\_association are the following:

- applied\_attribute\_name;
- assigned\_note;
- associated\_item;
- role.

#### 4.2.212.1 applied\_attribute\_name

The applied\_attribute\_name specifies the attribute to which the assigned Note (see 4.2.210) applies.

The applied\_attribute\_name need not be specified for a particular Note\_association.

#### 4.2.212.2 assigned\_note

The assigned\_note specifies the annotation.

See 4.3.1547 for the application assertion.

#### 4.2.212.3 associated\_item

The associated\_item specifies the item that is described by the associated annotation.

Each associated\_item may be one of the following: Activity (see 4.2.1), Activity\_method (see 4.2.3), Activity\_relationship (see 4.2.5), Alias\_identification (see 4.2.9), Alternate\_item\_relationship (see 4.2.11), Application\_context (see 4.2.22), Approval\_relationship (see 4.2.24), Assembly\_component\_relationship (see 4.2.26), Assembly\_substitute\_relationship (see 4.2.28), Cable\_pull\_information (see 4.2.33), Certification (see 4.2.38), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_attribute (see 4.2.47), Classification\_system (see 4.2.48), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_allocation (see 4.2.60), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Contract (see 4.2.63), Data\_element (see



4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_definition\_relationship (see 4.2.73), Data\_element\_specification (see 4.2.75), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_version (see 4.2.114), Drawing\_sheet (see 4.2.122), Drawing\_view (see 4.2.125), Effectivity (see 4.2.127), Effectivity\_relationship (see 4.2.129), Event\_reference (see 4.2.130), External\_library\_reference (see 4.2.132), Free\_segment (see 4.2.144), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), Functionality (see 4.2.155), General\_classification (see 4.2.156), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Offered\_function\_allocation (see 4.2.220), Organization\_relationship (see 4.2.225), Path (see 4.2.232), Path\_node (see 4.2.233), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Product\_class (see 4.2.263), Product\_class\_relationship (see 4.2.264), Product\_identification (see 4.2.268), Product\_specification (see 4.2.269), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Project\_relationship (see 4.2.272), Requirement (see 4.2.285), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Signal (see 4.2.309), Signal\_value (see 4.2.313), Specific\_item\_classification (see 4.2.321), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.1490, 4.3.1491, 4.3.1492, 4.3.1493, 4.3.1494, 4.3.1495, 4.3.1496, 4.3.1497, 4.3.1498, 4.3.1499, 4.3.1500, 4.3.1501, 4.3.1502, 4.3.1503, 4.3.1504, 4.3.1505, 4.3.1506, 4.3.1507, 4.3.1508, 4.3.1509, 4.3.1510, 4.3.1511, 4.3.1512, 4.3.1513, 4.3.1514, 4.3.1515, 4.3.1516, 4.3.1517, 4.3.1518, 4.3.1519, 4.3.1520, 4.3.1521, 4.3.1522, 4.3.1523, 4.3.1524, 4.3.1525, 4.3.1526, 4.3.1527, 4.3.1528, 4.3.1529, 4.3.1530, 4.3.1531, 4.3.1532, 4.3.1533, 4.3.1534, 4.3.1535, 4.3.1536, 4.3.1537, 4.3.1538, 4.3.1539, 4.3.1540, 4.3.1541, 4.3.1542, 4.3.1543, 4.3.1544, 4.3.1545, 4.3.1546, 4.3.1548, 4.3.1549, 4.3.1550, 4.3.1551, 4.3.1552, 4.3.1553, 4.3.1554, 4.3.1555, 4.3.1556, 4.3.1557, 4.3.1558, 4.3.1559, 4.3.1560, 4.3.1561, 4.3.1562, 4.3.1563, 4.3.1564, 4.3.1565, 4.3.1566, 4.3.1567, 4.3.1568, 4.3.1569, 4.3.1570, 4.3.1571, 4.3.1572, 4.3.1573, 4.3.1574, 4.3.1575, 4.3.1576, 4.3.1577, 4.3.1578, 4.3.1579, 4.3.1580, 4.3.1581, 4.3.1582, 4.3.1583, 4.3.1584, 4.3.1585, 4.3.1586, 4.3.1587, 4.3.1588, 4.3.1589, 4.3.1590, 4.3.1591, 4.3.1592, 4.3.1593, 4.3.1594, 4.3.1595, 4.3.1596, 4.3.1597, 4.3.1598, 4.3.1599, and 4.3.1600 for the application assertions.

#### 4.2.212.4 role

The role specifies the scope of the assigned annotation object.

## ISO 10303-212:2001(E)

The value of role is one of the following:

- attribute only;
- whole item.

NOTE 2 See 4.2.212.4.1 - 4.2.212.4.2 for the definition of each permissible value for role.

### 4.2.212.4.1 attribute only

attribute only: The associated annotation applies to the attribute specified through applied\_attribute\_name only.

### 4.2.212.4.2 whole item

whole item: The associated annotation applies to the whole item associated through associated\_item.

## 4.2.213 Note\_reference

A Note\_reference is a type of Cross\_reference (see 4.2.64) that is a reference made from one part of a diagram to another part between a note and the item being commented.

The data associated with a Note\_reference are the following:

- note\_presentation;
- presentation\_of\_item.

### 4.2.213.1 note\_presentation

The note\_presentation specifies the image that depicts the annotation in pictorial form.

See 4.3.1601 for the application assertion.

### 4.2.213.2 presentation\_of\_item

The presentation\_of\_item specifies the pictorial representation.

See 4.3.1602 for the application assertion.

### 4.2.214 Notification

A Notification is a report of the occurrence of an event.

NOTE A Notification can be caused by a Signal (see 4.2.309). In this case the Notification represents the information transmitted by this particular Signal (see 4.2.309).

EXAMPLE A Notification can be 'emergency switch-off', 'main drive on', 'Fresh-water reservoir empty', etc.

The data associated with a Notification are the following:

- description;
- id;
- language\_specification;
- text\_of\_message;
- trigger;
- version\_id.

#### 4.2.214.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Notification.

The description need not be specified for a particular Notification.

#### 4.2.214.2 id

The id specifies the identifier of the Notification.

#### 4.2.214.3 language\_specification

The language\_specification specifies a language spoken by human beings to communicate with each other verbally or in written form. This language is used to phrase the content of the 'text\_of\_message' attribute.

The language\_specification need not be specified for a particular Notification.

See 4.3.1603 for the application assertion.

#### 4.2.214.4 text\_of\_message

The text\_of\_message specifies the information conveyed by a Notification. The information shall be either for computer interpretation only, or for human interpretation.

The text\_of\_message need not be specified for a particular Notification.

#### **4.2.214.5 trigger**

The trigger specifies trigger event that initiates the notification.

Each trigger may be one of the following: Process\_variable (see 4.2.260), Signal (see 4.2.309), or Signal\_value (see 4.2.313).

See 4.3.1604, 4.3.1605, and 4.3.1606 for the application assertions.

#### **4.2.214.6 version\_id**

The version\_id specifies versioning information for the Notification.

The version\_id need not be specified for a particular Notification.

#### **4.2.215 Notification\_relationship**

A Notification\_relationship is a relation between two Notification (see 4.2.213) objects.

The data associated with a Notification\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

##### **4.2.215.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Notification\_relationship.

The description need not be specified for a particular Notification\_relationship.

##### **4.2.215.2 related**

The related specifies the second of the two Notification (see 4.2.213) objects related by the Notification\_relationship.

See 4.3.1607 for the application assertion.

##### **4.2.215.3 relating**

The relating specifies the first of the two Notification (see 4.2.213) objects related by the Notification\_relationship.

See 4.3.1608 for the application assertion.

##### **4.2.215.4 relation\_type**

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution;
- translation.

NOTE 1 See 4.2.215.4.1 - 4.2.215.4.5 for the definition of each predefined value for `relation_type`.

#### **4.2.215.4.1 alternate**

alternate: The `Notification_relationship` defines a relationship where the related Notification (see 4.2.213) is a possible substitute to the relating Notification (see 4.2.213).

NOTE 2 This concept refers to the possibility to replace the Notification (see 4.2.213). The actual replacement is addressed by 'substitution'.

#### **4.2.215.4.2 decomposition**

decomposition: The `Notification_relationship` defines a relationship where the related Notification (see 4.2.213) is one of the components into which the relating Notification (see 4.2.213) is divided.

#### **4.2.215.4.3 derivation**

derivation: The `Notification_relationship` defines a deriving relationship where the related Notification (see 4.2.213) is based on the relating Notification (see 4.2.213).

#### **4.2.215.4.4 substitution**

substitution: The `Notification_relationship` defines a relationship where the related Notification (see 4.2.213) replaces the relating Notification (see 4.2.213).

#### **4.2.215.4.5 translation**

translation: The `Notification_relationship` defines a relationship where the related Notification (see 4.2.213) is a transcription into another language of the relating Notification (see 4.2.213).

### **4.2.216 Numerical\_precision**

A `Numerical_precision` is the information about the mathematical precision of the numerical values contained in a graphical product model.

## ISO 10303-212:2001(E)

The data associated with a Numerical\_precision are the following:

- angular\_precision;
- distance\_precision.

### 4.2.216.1 angular\_precision

The angular\_precision specifies the maximum value for the absolute difference between two angle values when the creating system assumes these two angles to be identical.

### 4.2.216.2 distance\_precision

The distance\_precision specifies the maximum value for the absolute difference between two coordinate values when the creating system assumes these two coordinate values to be identical.

NOTE The distance\_precision is used to determine the tolerance range for continuous curve, surface connections, or for the determination of a point laying on a curve or surface.

### 4.2.217 Numerical\_value

A Numerical\_value is a type of Value\_with\_unit (see 4.2.360) that is an amount expressed as a multiple of a standardized quantity.

The data associated with a Numerical\_value are the following:

- value\_component.

### 4.2.217.1 value\_component

The value\_component specifies the numerical part of Numerical\_value.

### 4.2.218 Object\_designation

An Object\_designation is an identifier used to name physical or abstract items.

NOTE The Object\_designation allows to preserve the internal structure of the identification. By using the Object\_designation\_relationship (see 4.2.218) the structure of a reference designation can be expressed. Furthermore the information provided with Object\_designation enables a company that extends an existing plant to specify the identification for the added equipment consistently with the methodology used for the already existing items.

Each Object\_designation is either a Document\_designation (see 4.2.105), an Object\_reference\_designation (see 4.2.219), a Signal\_designation (see 4.2.310), or a Terminal\_designation (see 4.2.339).

The data associated with an Object\_designation are the following:

- designation\_system;
- designator;
- type\_of\_object\_designation.

#### **4.2.218.1 designation\_system**

The designation\_system specifies the methodology used to interpret the designator.

The designation\_system need not be specified for a particular Object\_designation.

See 4.3.1609 for the application assertion.

#### **4.2.218.2 designator**

The designator specifies an alphanumeric string identifying the physical or abstract item the Object\_designation is assigned to.

#### **4.2.218.3 type\_of\_object\_designation**

The type\_of\_object\_designation specifies the kind of the Object\_designation. The value is either user defined or predefined.

The type\_of\_object\_designation need not be specified for a particular Object\_designation. The type\_of\_object\_designation need not be specified for a particular Object\_designation.

The predefined value of type\_of\_object\_designation is one of the following:

- function designation;
- location designation;
- product designation.

NOTE See 4.2.218.3.1 - 4.2.218.3.3 for the definition of each predefined value for type\_of\_object\_designation.

##### **4.2.218.3.1 function designation**

function designation: The Object\_designation serves as an identifier or part of an identifier designating the functional aspect of a Function\_unit (see 4.2.148) or a Device (see 4.2.88) or a Location (see 4.2.192).

##### **4.2.218.3.2 location designation**

location designation: The Object\_designation serves as an identifier or as part of an identifier designating the place or position where the designated equipment is located.

#### 4.2.218.3.3 product designation

product designation: The Object\_designation serves as an identifier or part of an identifier designating the product aspect of a Function\_unit (see 4.2.148), a Device (see 4.2.88), or a Location (see 4.2.192).

### 4.2.219 Object\_designation\_relationship

An Object\_designation\_relationship is an association between Object\_designation (see 4.2.217) objects for the formation of hierarchically structured identifiers.

The data associated with an Object\_designation\_relationship are the following:

- related\_object\_designation;
- relating\_object\_designation.

#### 4.2.219.1 related\_object\_designation

The related\_object\_designation specifies the Object\_designation (see 4.2.217) objects that are the constituents of another Object\_designation (see 4.2.217).

See 4.3.1610 for the application assertion.

#### 4.2.219.2 relating\_object\_designation

The relating\_object\_designation specifies the Object\_designation (see 4.2.217) that is composed of one or more other Object\_designation (see 4.2.217) objects.

See 4.3.1611 for the application assertion.

### 4.2.220 Object\_reference\_designation

An Object\_reference\_designation is a type of Object\_designation (see 4.2.217) that is the name or identifier of a physical or abstract portion of an electrotechnical system. Within its scope, the Object\_reference\_designation is unique.

#### 4.2.221 Offered\_function\_allocation

An Offered\_function\_allocation is a relation that specifies a service that is accessible within a specific piece of equipment.



The data associated with a `Offered_function_allocation` are the following:

- `description`;
- `offered_functionality`;
- `performing_item`;
- `relation_type`.

#### **4.2.221.1 description**

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Offered_function_allocation`.

The `description` need not be specified for a particular `Offered_function_allocation`.

#### **4.2.221.2 offered\_functionality**

The `offered_functionality` specifies an available service.

See 4.3.1614 for the application assertion.

#### **4.2.221.3 performing\_item**

The `performing_item` specifies the equipment that provides the service.

Each `performing_item` may be one of the following: `Design_discipline_item_definition` (see 4.2.86), `Device` (see 4.2.88), `Physical_instance` (see 4.2.243), or `Product_component` (see 4.2.265).

See 4.3.1612, 4.3.1613, 4.3.1615, and 4.3.1616 for the application assertions.

#### **4.2.221.4 relation\_type**

The `relation_type` specifies the meaning of the relationship.

The value of `relation_type` is one of the following:

- `dedicated_function`;
- `offered_function`.

NOTE See 4.2.221.4.1 - 4.2.221.4.2 for the definition of each permissible value for `relation_type`.

##### **4.2.221.4.1 dedicated\_function**

`dedicated_function`: The service is no longer available i.e., the associated `Function_definition` (see 4.2.145) cannot be allocated further.

#### 4.2.221.4.2 offered function

offered function: The service is still available i.e., the associated Function\_definition (see 4.2.145) can still be allocated.

### 4.2.222 Operating\_temperature

An Operating\_temperature is the allowed ambient temperature of a component under normal operating conditions.

EXAMPLE The Operating\_temperature of a notebook computer ranges from 5°C to 35°C, whereas the Storage\_temperature (see 4.2.330) ranges from -20°C to 60°C.

The data associated with a Operating\_temperature are the following:

— temperature.

#### 4.2.222.1 temperature

The temperature specifies the value of the Operating\_temperature.

NOTE Minimum and maximum of Operating\_temperature may be specified by assigning a Value\_range (see 4.2.359) object.

See 4.3.1617 for the application assertion.

### 4.2.223 Ordinate\_dimension

An Ordinate\_dimension is a type of Dimension (see 4.2.93) that is the graphical presentation of a value of linear distance measure where the linear distance is parallel to an axis of the coordinate system of the item being dimensioned. The origin or datum of the linear distance dimension is a point, line, or plane surface corresponding to or coincident with an axis in the plane of the dimension and is perpendicular to the direction of measurement. Only the terminus of the dimension extent is indicated by a projection line parallel to the datum, the dimension value, and associated information.

The data associated with an Ordinate\_dimension are the following:

— component.

#### 4.2.223.1 component

The component specifies the projection line that shows the extension of the point, line, or surface.

See 4.3.1618 for the application assertion.

### 4.2.224 Organization

An Organization is a group of people involved in a particular business process.

The data associated with an Organization are the following:

- delivery\_address;
- id;
- name;
- organization\_type;
- postal\_address;
- visitor\_address.

#### **4.2.224.1 delivery\_address**

The delivery\_address specifies the address where to which goods are delivered.

The delivery\_address need not be specified for a particular Organization.

See 4.3.1619 for the application assertion.

#### **4.2.224.2 id**

The id specifies the identifier of the Organization.

**NOTE** The assignment of this attribute is usually controlled by a registration authority. The registration authority can be a public organization that assigns identifiers to corporations, or it can be the parent corporation that assigns component identifiers to its components.

**EXAMPLE** The id can be the code assigned to the Organization for a listing in a stock market, or it can be a department number.

#### **4.2.224.3 name**

The name specifies a speaking designation of the Organization.

#### **4.2.224.4 organization\_type**

The organization\_type specifies the type of the Organization. The value is either user defined or predefined.

The predefined value of organization\_type is one of the following:

- company;
- department;
- plant.

**NOTE** See 4.2.224.4.1 - 4.2.224.4.3 for the definition of each predefined value for organization\_type.

## **ISO 10303-212:2001(E)**

### **4.2.224.4.1 company**

company: Specifies that the Organization has the legal form of a business or enterprise.

### **4.2.224.4.2 department**

department: Specifies that the Organization is a division or office within a larger organization.

### **4.2.224.4.3 plant**

plant: Specifies that the Organization acts as a manufacturing site.

### **4.2.224.5 postal\_address**

The postal\_address specifies the address for letter mail.

The postal\_address need not be specified for a particular Organization.

See 4.3.1620 for the application assertion.

### **4.2.224.6 visitor\_address**

The visitor\_address specifies the address where the organization receives visitors.

The visitor\_address need not be specified for a particular Organization.

See 4.3.1621 for the application assertion.

### **4.2.225 Organization\_in\_contract**

An Organization\_in\_contract is a mechanism to associate the person who is signing a contract and the organization which the person is signing for, to a Contract (see 4.2.63).

The data associated with an Organization\_in\_contract are the following:

- contract;
- contracted\_organization;
- role\_of\_organization;
- signature.

#### **4.2.225.1 contract**

The contract specifies the Contract (see 4.2.63) in which the Organization\_in\_contract participates.

See 4.3.1622 for the application assertion.

#### **4.2.225.2 contracted\_organization**

The contracted\_organization specifies the organization that participates in the contract.

See 4.3.1624 for the application assertion.

### **4.2.225.3 role\_of\_organization**

The role\_of\_organization specifies the position of a signing organization with respect to the contract. The value is either user defined or predefined.

The predefined value of role\_of\_organization is one of the following:

— contractee;

— contractor.

NOTE See 4.2.225.3.1 - 4.2.225.3.2 for the definition of each predefined value for role\_of\_organization.

#### **4.2.225.3.1 contractee**

contractee: The tasks that are subject of the Contract (see 4.2.63) are assigned to the referenced Organization (see 4.2.223).

#### **4.2.225.3.2 contractor**

contractor: The referenced Organization (see 4.2.223) assigns the tasks that are subject of the Contract (see 4.2.63).

### **4.2.225.4 signature**

The signature specifies the Person (see 4.2.237) or Organization (see 4.2.223) who signed the contract on behalf of the contracted organization and the date of the signature.

See 4.3.1623 for the application assertion.

### **4.2.226 Organization\_relationship**

An Organization\_relationship is an association between two organizations.

## ISO 10303-212:2001(E)

EXAMPLE A team belongs to a department which itself belongs to a company. This organizational structure may be built using the Organization\_relationship mechanism.

The data associated with an Organization\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.226.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Organization\_relationship.

The description need not be specified for a particular Organization\_relationship.

### 4.2.226.2 related

The related specifies the other organization in an Organization\_relationship.

See 4.3.1625 for the application assertion.

### 4.2.226.3 relating

The relating specifies the one organization in an Organization\_relationship.

See 4.3.1626 for the application assertion.

### 4.2.226.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- hierarchy;
- legal succession;
- reorganization.

NOTE See 4.2.226.4.1 - 4.2.226.4.3 for the definition of each predefined value for relation\_type.

#### 4.2.226.4.1 hierarchy

hierarchy: The related Organization (see 4.2.223) is a sub organization of the relating Organization (see 4.2.223);

**4.2.226.4.2 legal succession**

legal succession: The related Organization (see 4.2.223) is the legal successor of the relating Organization (see 4.2.223).

NOTE An additional Date\_time\_assignment (see 4.2.80) may be applied to the Organization\_ - relationship in order to specify the time when the succession takes place.

**4.2.226.4.3 reorganization**

reorganization: The related Organization (see 4.2.223) is the successor of the relating Organization (see 4.2.223) due to an organizational transfer of responsibility.

EXAMPLE The name of a department changes due to reorganization.

**4.2.227 Outside\_diameter**

An Outside\_diameter is the outer width of a component with a body of circular cross section.

The data associated with an Outside\_diameter are the following:

— value\_of\_outside\_diameter.

**4.2.227.1 value\_of\_outside\_diameter**

The value\_of\_outside\_diameter specifies the value that indicates the outside diameter.

See 4.3.1627 for the application assertion.

**4.2.228 Page\_connector**

A Page\_connector is a cross reference used in schematic diagrams to indicate that a connecting line continues elsewhere on the same sheet or on another sheet of a drawing or in other related documents.

The data associated with a Page\_connector are the following:

— page\_connector\_id;

— part\_of;

— type\_of.

**4.2.228.1 page\_connector\_id**

The page\_connector\_id specifies an identifier for the Page\_connector that is unique within its scope.

**4.2.228.2 part\_of**

The part\_of specifies the Connecting\_line (see 4.2.58) of which the Page\_connector is a constituent.

See 4.3.1628 for the application assertion.

### 4.2.228.3 type\_of

The type\_of specifies the kind of the Page\_connector.

The value of type\_of is one of the following:

- central;
- exclusive;
- neutral.

NOTE See 4.2.228.3.1 - 4.2.228.3.3 for the definition of each permissible value for relation\_type.

#### 4.2.228.3.1 central

central: One or more other Page\_connector objects of type 'central' that are referred by this Page\_connector exist.

#### 4.2.228.3.2 exclusive

exclusive: There is exactly one other Page\_connector of type 'exclusive' referred by this particular Page\_connector.

#### 4.2.228.3.3 neutral

neutral: The counterpart the Page\_connector is referring to is not a part of the product data defined by this part of ISO 10303. One or more counterparts to a Page\_connector of type 'neutral' may exist.

### 4.2.229 Page\_connector\_presentation

A Page\_connector\_presentation is a visualization of a Page\_connector (see 4.2.227) in a schematic diagram.

The data associated with a Page\_connector\_presentation are the following:

- of\_page\_connector;
- type\_of.

#### 4.2.229.1 of\_page\_connector

The of\_page\_connector specifies the Page\_connector (see 4.2.227) object that is depicted by the Page\_connector\_presentation.

See 4.3.1629 for the application assertion.

#### 4.2.229.2 type\_of

The type\_of specifies the kind of the Page\_connector\_presentation.



The value of `type_of` is one of the following:

- `none`;
- `sink`;
- `source`.

NOTE See 4.2.229.2.1 - 4.2.229.2.3 for the definition of each permissible value for `type_of`.

#### 4.2.229.2.1 none

`none`: One or more other `Page_connector_presentation` objects of the type 'none' that are referred by the `Page_connector_presentation` exists.

#### 4.2.229.2.2 sink

`sink`: Exactly one `Page_connector_presentation` of the kind 'source' to which the `Page_connector_presentation` refers exists.

#### 4.2.229.2.3 source

`source`: One or more `Page_connector_presentation` objects of the kind 'sink' the `Page_connector_presentation` refers to exists.

### 4.2.230 Page\_connector\_reference

A `Page_connector_reference` is a type of `Cross_reference` (see 4.2.64) that is a reference made from one part of a diagram to another part between `Page_connector` (see 4.2.227) objects being part of the presentation of the same network.

The data associated with a `Page_connector_reference` are the following:

- `part_of`;
- `refers_to`.

#### 4.2.230.1 part\_of

The `part_of` specifies the `Page_connector_presentation` (see 4.2.228) to which the `Page_connector_reference` belongs.

See 4.3.1631 for the application assertion.

#### 4.2.230.2 refers\_to

The `refers_to` specifies the `Page_connector` (see 4.2.227) or `Page_connector_presentation` (see 4.2.228) object at which the `Page_connector_reference` points.

Each `refers_to` may be one of the following: `Page_connector` (see 4.2.227) or `Page_connector_presentation` (see 4.2.228).

See 4.3.1630 and 4.3.1632 for the application assertions.

### 4.2.231 Parallel\_dimension\_pair

A Parallel\_dimension\_pair is a type of Dimension\_sequence\_pair (see 4.2.97) that is the relationship between two dimensions of the same type, wherein their dimension lines are parallel and share a common baseline or datum.

### 4.2.232 Partial\_document\_assignment

A Partial\_document\_assignment is a type of Document\_assignment (see 4.2.102) that identifies a specific portion of the contents of a document or a set of documents and associates it to the product data in which this particular portion is relevant.

The data associated with a Partial\_document\_assignment are the following:

- document\_portion.

#### 4.2.232.1 document\_portion

The document\_portion specifies the word or group of words that convey the subject or sub contents of the Document (see 4.2.101).

EXAMPLE If the assigned document is a service manual of a complete power train, the 'document\_portion' attribute specifies that only the information related to electric fuses is relevant.

### 4.2.233 Path

A Path is a track taken to get from a starting-point to a destination. The Path may be specified as an ordered collection of Path\_segment (see 4.2.236) objects.

NOTE A Path is understood as a sequence of vertices and edges, where the edges are Path\_segment (see 4.2.236) objects, and the vertices are Path\_node (see 4.2.233) objects.

EXAMPLE A Path may specify the geometrical course of a cableway.

The data associated with a Path are the following:

- consists\_of;
- id;
- version\_id.

#### 4.2.233.1 consists\_of

The consists\_of specifies the Path\_segment (see 4.2.236) objects that define the geometrical or course of the Path. The associated Path\_segment (see 4.2.236) objects shall specify a continuous trail.

See 4.3.1633 for the application assertion.

**4.2.233.2 id**

The id specifies the identifier of the Path.

**4.2.233.3 version\_id**

The version\_id specifies versioning information for the Path.

The version\_id need not be specified for a particular Path.

**4.2.234 Path\_node**

A Path\_node is a vertex within a Path (see 4.2.232).

NOTE A Path (see 4.2.232) is understood as a sequence of vertices and edges, where the vertices are Path\_node objects.

The data associated with a Path\_node are the following:

- defined\_in;
- id;
- position.

**4.2.234.1 defined\_in**

The defined\_in specifies the Location (see 4.2.192) that provides the coordinate system used to define the position of the Path\_node.

See 4.3.1635 for the application assertion.

**4.2.234.2 id**

The id specifies the identifier of the Path\_node.

**4.2.234.3 position**

The position specifies the three-dimensional placement of the Path\_node.

See 4.3.1634 for the application assertion.

**4.2.235 Path\_node\_relationship**

A Path\_node\_relationship is the relation between two Path\_node (see 4.2.233) objects.

## ISO 10303-212:2001(E)

The data associated with an Path\_node\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.235.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Path\_node\_relationship.

The description need not be specified for a particular Path\_node\_relationship.

### 4.2.235.2 related

The related specifies the second of the two Path\_node (see 4.2.233) objects related by the Path\_node\_relationship.

See 4.3.1636 for the application assertion.

### 4.2.235.3 relating

The relating specifies the first of the two Path\_node (see 4.2.233) objects related by the Path\_node\_relationship.

See 4.3.1637 for the application assertion.

### 4.2.235.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- substitution.

NOTE 1 See 4.2.235.4.1 - 4.2.235.4.2 for the definition of each predefined value for relation\_type.

#### 4.2.235.4.1 alternate

alternate: The Path\_node\_relationship defines a relationship where the related Path\_node (see 4.2.233) is a possible substitute to the relating Path\_node (see 4.2.233).

NOTE 2 This concept refers to the possibility to replace the Path\_node (see 4.2.233). The actual replacement is addressed by 'substitution'.

**4.2.235.4.2 substitution**

substitution: The Path\_node\_relationship defines a relationship where the related Path\_node (see 4.2.233) replaces the relating Path\_node (see 4.2.233).

**4.2.236 Path\_relationship**

A Path\_relationship is the relation between two Path (see 4.2.232) objects.

The data associated with a Path\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

**4.2.236.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Path\_relationship.

The description need not be specified for a particular Path\_relationship.

**4.2.236.2 related**

The related specifies the second of the two Path (see 4.2.232) objects related by the Path\_relationship.

See 4.3.1638 for the application assertion.

**4.2.236.3 relating**

The relating specifies the first of the two Path (see 4.2.232) objects related by the Path\_relationship.

See 4.3.1639 for the application assertion.

**4.2.236.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- `alternate`;
- `derivation`;
- `substitution`.

NOTE 1 See 4.2.236.4.1 - 4.2.236.4.3 for the definition of each predefined value for `relation_type`.

### 4.2.236.4.1 `alternate`

`alternate`: The `Path_relationship` defines a relationship where the related Path (see 4.2.232) is a possible substitute to the relating Path (see 4.2.232).

NOTE 2 This concept refers to the possibility to replace the Path (see 4.2.232). The actual replacement is addressed by 'substitution'.

### 4.2.236.4.2 `derivation`

`derivation`: The `Path_relationship` defines a deriving relationship where the related Path (see 4.2.232) is based on the relating Path (see 4.2.232).

### 4.2.236.4.3 `substitution`

`substitution`: The `Path_relationship` defines a relationship where the related Path (see 4.2.232) replaces the relating Path (see 4.2.232).

## 4.2.237 `Path_segment`

A `Path_segment` is a continuous track between two `Path_node` (see 4.2.233) objects.

NOTE A Path (see 4.2.232) is understood as a sequence of vertices and edges, where the edges are `Path_segment` objects.

The data associated with an `Path_segment` are the following:

- `begins_at`;
- `defined_in`;
- `ends_at`;
- `form`;
- `id`.

### 4.2.237.1 `begins_at`

The `begins_at` specifies the starting point of the `Path_segment`.

See 4.3.1642 for the application assertion.

**4.2.237.2 defined\_in**

The `defined_in` specifies the Location (see 4.2.192) that provides the coordinate system to define the `Path_segment`.

See 4.3.1641 for the application assertion.

**4.2.237.3 ends\_at**

The `ends_at` specifies the end point of the `Path_segment`.

See 4.3.1643 for the application assertion.

**4.2.237.4 form**

The `form` specifies a three-dimensional curve that defines the shape of the `Path_segment`. This curve shall be defined in the coordinate system specified by the Location (see 4.2.192) associated through `defined_in`.

See 4.3.1640 for the application assertion.

**4.2.237.5 id**

The `id` specifies the identifier of the `Path_segment`.

**4.2.238 Person**

A Person is an individual human being who has some relationship to the product data. The Person shall always be identified in the context of one or more organizations.

The data associated with a Person are the following:

- `first_name`;
- `last_name`;
- `middle_names`;
- `preferred_business_address`;
- `prefix_titles`;
- `suffix_titles`.

**4.2.238.1 first\_name**

The `first_name` specifies the first element in the list of a person's list of forenames.

The `first_name` need not be specified for a particular Person.

#### **4.2.238.2 last\_name**

The last\_name specifies the person's surname.

The last\_name need not be specified for a particular Person.

#### **4.2.238.3 middle\_names**

The middle\_names specifies the other person's forenames, if there are any.

There may be one or more middle\_names for a Person. The middle\_names need not be specified for a particular Person.

#### **4.2.238.4 preferred\_business\_address**

The preferred\_business\_address specifies the location of the office of the Person.

The preferred\_business\_address need not be specified for a particular Person.

See 4.3.1644 for the application assertion.

#### **4.2.238.5 prefix\_titles**

The prefix\_titles specifies the word, or group of words, that specify the person's social or professional standing and appear before his or her name.

There may be one or more prefix\_titles for a Person. The prefix\_titles need not be specified for a particular Person.

#### **4.2.238.6 suffix\_titles**

The suffix\_titles specifies the word, or group of words, that specify the person's social or professional standing and appear after his or her name.

There may be one or more suffix\_titles for a Person. The suffix\_titles need not be specified for a particular Person.

#### **4.2.239 Person\_in\_organization**

A Person\_in\_organization is the specification of a Person (see 4.2.237) in the context of an Organization (see 4.2.223).



The data associated with an `Person_in_organization` are the following:

- `associated_organization`;
- `associated_person`;
- `id`;
- `location`;
- `role`.

#### **4.2.239.1 associated\_organization**

The `associated_organization` specifies the Organization (see 4.2.223) with which the Person (see 4.2.237) is associated.

See 4.3.1646 for the application assertion.

#### **4.2.239.2 associated\_person**

The `associated_person` specifies the Person (see 4.2.237).

See 4.3.1647 for the application assertion.

#### **4.2.239.3 id**

The `id` specifies an identifier of the person. The identifier shall be unique within the scope of the '`associated_organization`'.

EXAMPLE The `id` may be a staff number or a user `id` in a computer system.

#### **4.2.239.4 location**

The `location` specifies the relevant address of the `Person_in_organization`.

The `location` need not be specified for a particular `Person_in_organization`.

See 4.3.1645 for the application assertion.

#### **4.2.239.5 role**

The `role` specifies the relationship between the Person (see 4.2.237) and the Organization (see 4.2.223).

#### **4.2.240 Person\_in\_organization\_relationship**

A `Person_in_organization_relationship` is a mechanism which allows to specify an relationship between two persons in an organization.

## ISO 10303-212:2001(E)

EXAMPLE The owner of important product data leaves a company and therefore has a successor. This relationship may be built up using the `Person_in_organization_relationship` mechanism.

The data associated with an `Person_in_organization_relationship` are the following:

- `description`;
- `related`;
- `relating`;
- `relation_type`.

### 4.2.240.1 description

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Person_in_organization_relationship`.

The `description` need not be specified for a particular `Person_in_organization_relationship`.

### 4.2.240.2 related

The `related` specifies the second of the two objects related by the `Person_in_organization_relationship`.

See 4.3.1648 for the application assertion.

### 4.2.240.3 relating

The `relating` specifies the first of the two objects related by the `Person_in_organization_relationship`.

See 4.3.1649 for the application assertion.

### 4.2.240.4 relation\_type

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of `relation_type` is one of the following:

- `successor`.

NOTE See 4.2.240.4.1 for the definition of each predefined value for `relation_type`.

#### 4.2.240.4.1 successor

`successor`: The related `Person_in_organization` (see 4.2.238) is the successor of the relating `Person_in_organization` (see 4.2.238).

### 4.2.241 Person\_organization\_assignment

A Person\_organization\_assignment is a relation that associates an Organization (see 4.2.223) or a Person\_in\_organization (see 4.2.238) with an item.

The data associated with a Person\_organization\_assignment are the following:

- assigned\_person\_or\_organization;
- description;
- is\_applied\_to;
- role.

#### 4.2.241.1 assigned\_person\_or\_organization

The assigned\_person\_or\_organization specifies the concerned individual or organization.

Each assigned\_person\_or\_organization may be one of the following: Organization (see 4.2.223) or Person\_in\_organization (see 4.2.238).

See 4.3.1724 and 4.3.1730 for the application assertions.

#### 4.2.241.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Person\_organization\_assignment.

The description need not be specified for a particular Person\_organization\_assignment.

#### 4.2.241.2 is\_applied\_to

The is\_applied\_to specifies the item with which the Person\_organization\_assignment is associated.

Each is\_applied\_to may be one of the following: Activity (see 4.2.1), Activity\_element (see 4.2.2), Activity\_method\_assignment (see 4.2.4), Activity\_relationship (see 4.2.5), Alternate\_item\_relationship (see 4.2.11), Approval\_status (see 4.2.25), Assembly\_component\_relationship (see 4.2.26), Assembly\_substitute\_relationship (see 4.2.28), Cable\_pull\_information (see 4.2.33), Certification (see 4.2.38), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_association (see 4.2.46), Classification\_attribute (see 4.2.47), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_allocation (see 4.2.60), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Contract (see 4.2.63), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Data\_element\_specification (see 4.2.75), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see

## ISO 10303-212:2001(E)

4.2.124), Free\_segment (see 4.2.144), Function\_definition (see 4.2.145), Function\_definition - relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function - unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition - relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), Functionality (see 4.2.155), General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface - port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Organization\_relationship (see 4.2.225), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Person\_in\_organization (see 4.2.238), Person\_in\_organization\_relationship (see 4.2.239), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Port\_association (see 4.2.249), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process - variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Requirement\_document\_assignment (see 4.2.287), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section - interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.1650, 4.3.1651, 4.3.1652, 4.3.1653, 4.3.1654, 4.3.1655, 4.3.1656, 4.3.1657, 4.3.1658, 4.3.1659, 4.3.1660, 4.3.1661, 4.3.1662, 4.3.1663, 4.3.1664, 4.3.1665, 4.3.1666, 4.3.1667, 4.3.1668, 4.3.1669, 4.3.1670, 4.3.1671, 4.3.1672, 4.3.1673, 4.3.1674, 4.3.1675, 4.3.1676, 4.3.1677, 4.3.1678, 4.3.1679, 4.3.1680, 4.3.1681, 4.3.1682, 4.3.1683, 4.3.1684, 4.3.1685, 4.3.1686, 4.3.1687, 4.3.1688, 4.3.1689, 4.3.1690, 4.3.1691, 4.3.1692, 4.3.1693, 4.3.1694, 4.3.1695, 4.3.1696, 4.3.1697, 4.3.1698, 4.3.1699, 4.3.1700, 4.3.1701, 4.3.1702, 4.3.1703, 4.3.1704, 4.3.1705, 4.3.1706, 4.3.1707, 4.3.1708, 4.3.1709, 4.3.1710, 4.3.1711, 4.3.1712, 4.3.1713, 4.3.1714, 4.3.1715, 4.3.1716, 4.3.1717, 4.3.1718, 4.3.1719, 4.3.1720, 4.3.1721, 4.3.1722, 4.3.1723, 4.3.1725, 4.3.1726, 4.3.1727, 4.3.1728, 4.3.1729, 4.3.1731, 4.3.1732, 4.3.1733, 4.3.1734, 4.3.1735, 4.3.1736, 4.3.1737, 4.3.1738, 4.3.1739, 4.3.1740, 4.3.1741, 4.3.1742, 4.3.1743, 4.3.1744, 4.3.1745, 4.3.1746, 4.3.1747, 4.3.1748, 4.3.1749, 4.3.1750, 4.3.1751, 4.3.1752, 4.3.1753, 4.3.1754, 4.3.1755, 4.3.1756, 4.3.1757, 4.3.1758, 4.3.1759, 4.3.1760, 4.3.1761, 4.3.1762, 4.3.1763, 4.3.1764, 4.3.1765, 4.3.1766, 4.3.1767, 4.3.1768, and 4.3.1769 for the application assertions.

### 4.2.241.3 role

The role specifies the responsibility of the assigned individual or organization with respect to the item to which it is applied. The value is either user defined or predefined.

The predefined value of role is one of the following:

- author;
- classification officer;
- creator;
- custodian;
- customer;
- design supplier;
- editor;
- id owner;
- inspector;
- local representative;
- location;
- manufacturer;
- operator;
- owner;
- scope;
- supplier;
- wholesaler.

NOTE See 4.2.241.3.1 - 4.2.241.3.17 for the definition of each predefined value for role.

#### **4.2.241.3.1 author**

author: The referenced item has been originated by the individual or organization.

#### **4.2.241.3.2 classification officer**

classification officer: The assigned person or organization is formally responsible for the classification of the referenced object;.

#### **4.2.241.3.3 creator**

creator: The referenced item has been created by the individual or organization.

## **ISO 10303-212:2001(E)**

### **4.2.241.3.4 custodian**

custodian: The assigned individual or organization is responsible for the existence and integrity of the referenced item.

### **4.2.241.3.5 customer**

customer: The assigned individual or organization acts as a purchaser or consumer of the referenced item.

NOTE The customer may be part of the same organization as the supplier.

### **4.2.241.3.6 design supplier**

design supplier: The assigned Person (see 4.2.237) or Organization (see 4.2.223) is the one who delivers the data describing the referenced object.

### **4.2.241.3.7 editor**

editor: The assigned individual or organization is responsible for making changes of the referenced item.

### **4.2.241.3.8 id owner**

id owner: The assigned Person (see 4.2.237) or Organization (see 4.2.223) is responsible for the designation of an identifier.

### **4.2.241.3.9 inspector**

inspector: The task of the assigned individual or organization is to supervise the referenced item and to make reports.

### **4.2.241.3.10 local representative**

local representative: The assigned individual or organization acts as a local contact point for the referenced item.

EXAMPLE The jobsite management of a construction site may act as local representative of its company.

### **4.2.241.3.11 location**

location: The assigned Organization (see 4.2.223) is the place where the referenced object can be found or where it takes place.

### **4.2.241.3.12 manufacturer**

manufacturer: The assigned individual or organization produces the referenced item.

### **4.2.241.3.13 operator**

operator: The assigned individual or organization is running the referenced item.

**4.2.241.3.14 owner**

owner: The assigned individual or organization owns the referenced item.

**4.2.241.3.15 scope**

scope: The assigned individual or organization specifies the range of validity for the referenced item.

**4.2.241.3.16 supplier**

supplier: The assigned individual or organization provides the referenced item.

**4.2.241.3.17 wholesaler**

wholesaler: The assigned Person (see 4.2.237) or Organization (see 4.2.223) is the one who is in the sales chain between the manufacturer and the supplier.

**4.2.242 Physical\_assembly\_relationship**

A `Physical_assembly_relationship` is the mechanism to relate one `Physical_instance` (see 4.2.243) as a component to another `Physical_instance` (see 4.2.243) that plays the role of an assembly.

The data associated with a `Physical_assembly_relationship` are the following:

- `is_realization_of`;
- `physical_assembly`;
- `physical_component`.

**4.2.242.1 is\_realization\_of**

The `is_realization_of` specifies the `Device` (see 4.2.88) the of which the physical component is an occurrence.

See 4.3.1770 for the application assertion.

**4.2.242.2 physical\_assembly**

The `physical_assembly` specifies the `Physical_instance` (see 4.2.243) that serves as the assembly in the physical structure.

See 4.3.1771 for the application assertion.

**4.2.242.3 physical\_component**

The `physical_component` specifies the `Physical_instance` (see 4.2.243) that serves as a component in the physical structure.

See 4.3.1772 for the application assertion.

**4.2.243 Physical\_document**

## ISO 10303-212:2001(E)

A `Physical_document` is a type of `Document_representation` (see 4.2.110) that is a piece of product data that is archived in non-digital form.

EXAMPLE Paper plots of technical drawings, microfiche, or paper documents, such as calculations or test reports, are examples for a `Physical_document`.

The data associated with a `Physical_document` are the following:

- `component`.

### 4.2.243.1 component

The `component` is any portion of a `Physical_document`.

See 4.3.1773 for the application assertion.

### 4.2.244 Physical\_instance

A `Physical_instance` is a type of `Product_constituent` (see 4.2.266) that is the denomination of a physically realized item that is identified by a lot id or by a serial number.

The data associated with a `Physical_instance` are the following:

- `description`;
- `inventory_number`;
- `is_realization_of`;
- `lot_id`;
- `serial_number`.

#### 4.2.244.1 description

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Physical_instance`.

The `description` need not be specified for a particular `Physical_instance`.

#### 4.2.244.2 inventory\_number

The `inventory_number` specifies an alphanumerical string to identify an item in the detailed list of articles, such as goods and chattels, found to be in the possession of a person or enterprise.

The `inventory_number` need not be specified for a particular `Physical_instance`.

#### 4.2.244.3 is\_realization\_of

The `is_realization_of` specifies the item that comprises the information defining the `Physical_instance`.



NOTE This information is only necessary for the 'root' component in a physical assembly. For a component this information is already specified through `Physical_assembly_relationship` (see 4.2.241) which references the defining item through the corresponding `Device` (see 4.2.88).

Each `is_realization_of` may be one of the following: `Design_discipline_item_definition` (see 4.2.86) or `Product_identification` (see 4.2.268).

See 4.3.1774 and 4.3.1775 for the application assertions.

#### 4.2.244.4 `lot_id`

The `lot_id` specifies the identifier of the lot of which the `Physical_instance` is a part.

The `lot_id` need not be specified for a particular `Physical_instance`.

#### 4.2.244.5 `serial_number`

The `serial_number` specifies an alphanumerical string to identify a piece of equipment uniquely within a production series. It is indelibly attached to the equipment.

EXAMPLE In the case of the replacement of a device due to a repair, the new device will have another serial number indicating that a replacement of the device has taken place.

The `serial_number` need not be specified for a particular `Physical_instance`.

#### 4.2.245 `Physical_model`

A `Physical_model` is a type of `Document_representation` (see 4.2.110) that is a model of the layout of a complete system or some portion thereof made from materials such as clay or wood in a specific scale.

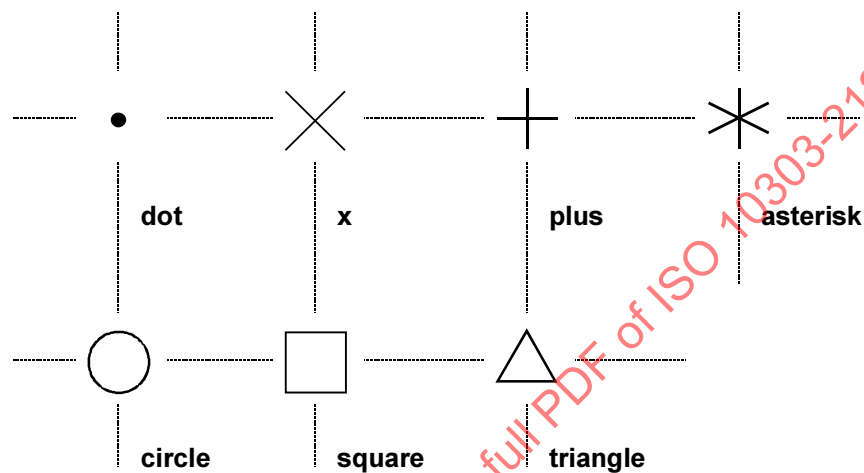
EXAMPLE A `Physical_model` is used to give the contractor an impression of the arrangement of switchgears and transformers within a planned power plant.

#### 4.2.246 `Point_2d`

A `Point_2d` is a position in a two-dimensional cartesian coordinate space.

#### 4.2.247 `Point_marker_symbol`

A `Point_marker_symbol` is a type of `Predefined_symbol` (see 4.2.255) that is used to visually present the location of a point in a drawing sheet, drawing view, or another symbol.



**Figure 16 - Predefined point marker symbols**

The data associated with a `Point_marker_symbol` are the following:

— `symbol_type`.

#### 4.2.247.1 `symbol_type`

The `symbol_type` specifies an alphanumeric string identifying the `Point_marker_symbol`.

The value of symbol\_type is one of the following:

- asterisk;
- circle;
- dot;
- plus;
- square;
- triangle;
- x.

NOTE See 4.2.247.1.1 - 4.2.247.1.7 for the definition of each permissible value for symbol\_type.

#### 4.2.247.1.1 asterisk

asterisk: the origin of the symbol is the geometrical centre of the asterisk. The size and graphical representation of the symbol are shown in Figure 16. The reference point lies on the intersection of the dashed lines.

#### 4.2.247.1.2 circle

circle: the origin of the symbol is the centre. The size and graphical representation of the symbol are shown in Figure 16. The reference point lies on the intersection of the dashed lines.

#### 4.2.247.1.3 dot

dot: a dot symbol is depicted as a circle with a fill pattern applied to it. The origin of the dot symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 16. The reference point lies on the intersection of the dashed lines.

#### 4.2.247.1.4 plus

plus: a plus symbol is depicted as two perpendicular line segments. The origin of the symbol is the intersection point of the two line segments. The size and graphical representation of the symbol are shown in Figure 16. The reference point lies on the intersection of the dashed lines.

#### 4.2.247.1.5 square

square: a square symbol is depicted as an even-sided rectangle. The origin of the symbol is the geometrical centre of the square. The size and graphical representation of the symbol are shown in Figure 16. The reference point lies on the intersection of the dashed lines.

#### 4.2.247.1.6 triangle

triangle: the origin of the symbol is the geometrical centre of the triangle. The size and graphical representation of the symbol are shown in Figure 16. The reference point lies on the intersection of the dashed lines.

**4.2.247.1.7 x**

x: the origin of the symbol is the intersection point of the two line segments. The size and graphical representation of the symbol are shown in Figure 16. The reference point lies on the intersection of the dashed lines.

**4.2.248 Port**

A Port is the occurrence of an Interface\_port (see 4.2.171) used to access a functional module.

The data associated with a Port are the following:

- associated\_interface\_port;
- description;
- extended\_designation;
- id;
- port\_of.

**4.2.248.1 associated\_interface\_port**

The associated\_interface\_port specifies the Interface\_port (see 4.2.171) that characterizes the access to the functionality of the accessed functional unit.

See 4.3.1776 for the application assertion.

**4.2.248.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Port.

The description need not be specified for a particular Port.

**4.2.248.2 extended\_designation**

The extended\_designation specifies a structured label for the Port.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

The extended\_designation need not be specified for a particular Port.

See 4.3.1778 for the application assertion.

**4.2.248.3 id**

The id specifies the identifier for the Port.

#### 4.2.248.4 port\_of

The port\_of specifies the Single\_function\_unit (see 4.2.315) to which the Port belongs.

See 4.3.1777 for the application assertion.

#### 4.2.249 Port\_allocation

A Port\_allocation is the association of a Port (see 4.2.247) object and its implementation.

NOTE A Port (see 4.2.247) may be implemented through a Function\_unit (see 4.2.148).

The data associated with a Port\_allocation are the following:

- allocated\_port;
- description;
- item\_allocation;
- port\_implementation.

##### 4.2.249.1 allocated\_port

The allocated\_port specifies the Port (see 4.2.247) object that is subject of implementation.

See 4.3.1780 for the application assertion.

##### 4.2.249.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Port\_allocation.

The description need not be specified for a particular Port\_allocation.

##### 4.2.249.3 item\_allocation

The item\_allocation specifies the Functional\_unit\_allocation (see 4.2.154) object that allocates the Function\_unit (see 4.2.148) which owns the ports to its implementation.

The item\_allocation need not be specified for a particular Port\_allocation.

See 4.3.1779 for the application assertion.

##### 4.2.249.4 port\_implementation

The port\_implementation specifies the items that implement the Port (see 4.2.247).

See 4.3.1781 for the application assertion.

## 4.2.250 Port\_association

A Port\_association is the relation between an Interface\_port (see 4.2.171) and a Port (see 4.2.247). In a hierarchical functional decomposition, the Port\_association specifies the Port (see 4.2.247) of which the higher level Interface\_port (see 4.2.171) is an abstraction.

The data associated with a Port\_association are the following:

- associated\_interface\_port;
- associated\_port.

### 4.2.250.1 associated\_interface\_port

The associated\_interface\_port specifies the higher level Interface\_port (see 4.2.171).

See 4.3.1782 for the application assertion.

### 4.2.250.2 associated\_port

The associated\_port specifies the Port (see 4.2.247) to which the Interface\_port (see 4.2.171) is associated.

See 4.3.1783 for the application assertion.

## 4.2.251 Port\_relationship

A Port\_relationship is the relation between two Port (see 4.2.247) objects.

The data associated with a Port\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.251.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Port\_relationship.

The description need not be specified for a particular Port\_relationship.

### 4.2.251.2 related

The related specifies the second of the two Port (see 4.2.247) objects related by the Port\_relationship.

See 4.3.1784 for the application assertion.

### 4.2.251.3 relating

The relating specifies the first of the two Port (see 4.2.247) objects related by the Port\_relationship.

See 4.3.1785 for the application assertion.

### 4.2.251.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- decomposition;
- redundancy.

NOTE See 4.2.251.4.1 and 4.2.251.4.2 for the definition of each predefined value for relation\_type.

#### 4.2.251.4.1 decomposition

decomposition: The Port\_relationship defines a relationship where the related Port (see 4.2.247) is one of the components into which the relating Port (see 4.2.247) is broken down.

#### 4.2.251.4.2 redundancy

redundancy: The Port\_relationship defines a relationship where the related Port (see 4.2.247) is replicated by the relating Port (see 4.2.247).

EXAMPLE To provide for a fail-safe service a Port (see 4.2.247) is replicated. If one Port (see 4.2.247) fails, the other is still in service.

### 4.2.252 Predefined\_colour

A Predefined\_colour is a type of Colour (see 4.2.50) that shall be supported by implementations of this part of ISO 10303. A Predefined\_colour is defined by an explicit listing of the proportions of blue, green, and red colour according to Table 1.

**Table 1 - RGB values for predefined colours**

Colour name	Red	Green	Blue
Colour	0.0	0.0	0.0
Red	1.0	0.0	0.0
Green	0.0	1.0	0.0
Blue	0.0	0.0	1.0
Yellow	1.0	1.0	0.0
Magenta	1.0	0.0	1.0
Cyan	0.0	1.0	1.0
White	1.0	1.0	1.0

The data associated with a `Predefined_colour` are the following:

— `colour_id`.

#### **4.2.252.1 colour\_id**

The `colour_id` specifies an alphanumerical string identifying the `Predefined_colour` in accordance to the definitions given above.

#### **4.2.253 Predefined\_connection**

A `Predefined_connection` is a type of `Connection` (see 4.2.59) that shall not be modified.

**NOTE** Changes of existing connections can occur when the wiring of an electrotechnical system is modified due to design changes or maintenance activities.

**EXAMPLE** In an existing electrotechnical system, changes occur due to additional customer requirements. These changes affect the connectivity within the system. This particular system contains connections that must not be modified because these connections were carefully designed to avoid electromagnetic interference. To prevent those critical connections from being redesigned, they are identified as `Predefined_connection` objects.

#### **4.2.254 Predefined\_data\_element**

A `Predefined_data_element` is a type of `Data_element` (see 4.2.70) that is completely specified in this part of ISO 10303. In a tree of `Data_element` (see 4.2.70) objects formed by using the `Data_element_` - relationship (see 4.2.74), `Predefined_data_element` objects shall occur only as leaves.



The data associated with a `Predefined_data_element` are the following:

— data.

#### 4.2.254.1 data

The data specifies the facts that are represented by the `Predefined_data_element`.

Each data may be one of the following: `Body_breadth` (see 4.2.30), `Body_height` (see 4.2.31), `Body_length` (see 4.2.32), `Component_colour` (see 4.2.53), `Cross_section` (see 4.2.65), `Mass` (see 4.2.200), `Material` (see 4.2.201), `Mounting_features` (see 4.2.203), `Operating_temperature` (see 4.2.221), `Outside_diameter` (see 4.2.226), `Rated_current` (see 4.2.278), `Rated_power` (see 4.2.279), `Rated_voltage` (see 4.2.280), or `Storage_temperature` (see 4.2.330).

See 4.3.1786, 4.3.1787, 4.3.1788, 4.3.1789, 4.3.1790, 4.3.1791, 4.3.1792, 4.3.1793, 4.3.1794, 4.3.1795, 4.3.1796, 4.3.1797, 4.3.1798, and 4.3.1799 for the application assertions.

#### 4.2.255 Predefined\_line\_font

A `Predefined_line_font` is a type of `Line_font` (see 4.2.189) that has a specific physical appearance as defined in this part of ISO 10303. Table 2 gives the length of each line segment and space, in millimetres.

**Table 2 - Line segment and space lengths for predefined curve font lists**

Curve pattern name	Segment (mm)	Space (mm)	Segment (mm)	Space (mm)	Segment (mm)	Space (mm)	Number of Segments
Curvenuous							0
Dashed	4.0	1.5					2
Chain	7.0	1.0	1.0	1.0			4
Chain double dash	7.0	1.0	1.0	1.0	1.0	1.0	6
Dotted	1.0	1.0					2

The data associated with a `Predefined_line_font` are the following:

— font\_id.

#### 4.2.255.1 font\_id

The `font_id` specifies an alphanumerical string identifying the `Predefined_line_font` in accordance to the definitions given in Table 2.

#### 4.2.256 Predefined\_symbol

## ISO 10303-212:2001(E)

A `Predefined_symbol` is a type of `Annotation_symbol` (see 4.2.20) that has a specific physical appearance as defined in this part of ISO 10303.

Each `Predefined_symbol` is either a `Dimension_symbol` (see 4.2.98), a `Geometrical_tolerance_symbol` (see 4.2.161), a `Point_marker_symbol` (see 4.2.246), or a `Terminator_symbol` (see 4.2.341).

### 4.2.257 `Predefined_text_font`

A `Predefined_text_font` is a type of `Text_font` (see 4.2.344) that has an appearance and a character code in accordance to the referenced standard.

The data associated with a `Predefined_text_font` are the following:

— `font_id`.

#### 4.2.257.1 `font_id`

The `font_id` is a string identifying the standard that defines the `Predefined_text_font`.

The value of `font_id` is one of the following:

— `iec 61286`;

— `iso 646`;

— `iso 3098-1`

— `iso 6937`;

— `iso 8859-1`;

— `iso 10646`.

NOTE See 4.2.257.1.1 – 4.2.257.1.5 for the definition of each predefined value for `character_code`.

##### 4.2.257.1.1 `iec 61286`

`iec 61286`: The coded character set used to encode the data is in accordance to IEC 61286.

##### 4.2.257.1.2 `iso 646`

`iso 646`: The coded character set used to encode the data is in accordance to the International Reference Version (IRV) of ISO/IEC 646.

NOTE The IRV of ISO/IEC 646 is identical with the character reservoir commonly known as ASCII.

##### 4.2.257.1.3 `iso 3098-1`

`iso 3098-1`: The coded character set used to encode the data is in accordance to ISO 3098-1.

##### 4.2.257.1.4 `iso 6937`

iso 6937: The coded character set used to encode the data is in accordance to ISO/IEC 6937.

#### **4.2.257.1.5 iso 8859-1**

iso 8859-1: The coded character set used to encode the data is in accordance to ISO 8859-1.

#### **4.2.257.1.6 iso 10646**

iso 10646: The coded character set used to encode the data is in accordance to ISO/IEC 10646.

### **4.2.258 Preferred\_equipment\_assignment**

A Preferred\_equipment\_assignment is the association of the type of equipment to the Signal (see 4.2.309), that is the recommended equipment to process or transmit the Signal (see 4.2.309).

The data associated with a Preferred\_equipment\_assignment are the following:

- preferred\_equipment;
- related\_signal;
- valid\_context.

#### **4.2.258.1 preferred\_equipment**

The preferred\_equipment specifies the favoured equipment.

Each preferred\_equipment may be one of the following: Design\_discipline\_item\_definition (see 4.2.86) or Function\_definition (see 4.2.145).

See 4.3.1800 and 4.3.1801 for the application assertions.

#### **4.2.258.2 related\_signal**

The related\_signal specifies the Signal (see 4.2.309) that is to be processed.

See 4.3.1804 for the application assertion.

#### **4.2.258.3 valid\_context**

The valid\_context specifies circumstances under which the recommendation specified through the preferred\_equipment attribute is valid.

The valid\_context need not be specified for a particular Preferred\_equipment\_assignment.

Each valid\_context may be one of the following: Organization (see 4.2.223) or Product\_class (see 4.2.263).

See 4.3.1802 and 4.3.1803 for the application assertions.

### 4.2.259 Preferred\_item\_allocation

The Preferred\_item\_allocation is the association of those items to a Function\_unit (see 4.2.148) that are the favoured realization of that Function\_unit (see 4.2.148).

EXAMPLE In a mining environment, only explosion-proof equipment may be used.

The data associated with a Preferred\_item\_allocation are the following:

- description;
- functional\_definition;
- preferred\_item.

#### 4.2.259.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Preferred\_item\_allocation.

The description need not be specified for a particular Preferred\_item\_allocation.

#### 4.2.259.2 functional\_definition

The functional\_definition specifies the Function\_unit (see 4.2.148) object.

See 4.3.1807 for the application assertion.

#### 4.2.259.3 preferred\_item

The preferred\_item specifies the item that represents the favoured implementation of the Function\_unit (see 4.2.148) object.

Each preferred\_item may be one of the following: Design\_discipline\_item\_definition (see 4.2.86) or Function\_definition (see 4.2.145).

See 4.3.1805 and 4.3.1806 for the application assertions.

### 4.2.260 Preferred\_item\_terminal\_allocation

A Preferred\_item\_terminal\_allocation specifies the favoured implementation of the access nodes of a functional unit.

The data associated with a Preferred\_item\_terminal\_allocation are the following:

- description;
- functional\_definition;
- item\_allocation;
- preferred\_node.

#### **4.2.260.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Preferred\_item\_terminal\_allocation.

The description need not be specified for a particular Preferred\_item\_terminal\_allocation.

#### **4.2.260.2 functional\_definition**

The functional\_definition specifies the allocated Port (see 4.2.247) objects.

See 4.3.1810 for the application assertion.

#### **4.2.260.3 item\_allocation**

The item\_allocation specifies the Preferred\_item\_allocation (see 4.2.258) object that allocates the Function\_unit (see 4.2.148), which owns the ports, to its implementation.

See 4.3.1811 for the application assertion.

#### **4.2.260.4 preferred\_node**

The preferred\_node specifies the access nodes that are the preferred implementation of the Port (see 4.2.247) objects specified by the 'functional\_definition' attribute.

Each preferred\_node may be one of the following: Interface\_port (see 4.2.171) or Interface\_terminal (see 4.2.174).

See 4.3.1808 and 4.3.1809 for the application assertions.

#### **4.2.261 Process\_variable**

A Process\_variable is a parameter used to control or monitor a process.

## ISO 10303-212:2001(E)

**NOTE** To be processed by an electrotechnical system a `Process_variable` needs to be converted into a `Signal` (see 4.2.309). `Process_variable` objects are considered to be outside of the electrotechnical system. They may serve as an input, output or control of the system.

**EXAMPLE** `Process_variable` objects are parameters such as rotational speed, temperature, or pressure.

The data associated with a `Process_variable` are the following:

- `associated_value`;
- `description`;
- `id`.

### 4.2.261.1 `associated_value`

The `associated_value` specifies the quantity that is assigned to the `Process_variable`.

The `associated_value` need not be specified for a particular `Process_variable`.

See 4.3.1812 for the application assertion.

### 4.2.261.2 `description`

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Process_variable`.

The `description` need not be specified for a particular `Process_variable`.

### 4.2.261.3 `id`

The `id` specifies the identifier of the `Process_variable`.

## 4.2.262 `Process_variable_relationship`

A `Process_variable_relationship` is the relation between two `Process_variable` (see 4.2.260) objects.

The data associated with a `Process_variable_relationship` are the following:

- `description`;
- `related`;
- `relating`;
- `relation_type`.

### 4.2.262.1 `description`

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the `Process_variable_relationship`.

The description need not be specified for a particular Process\_variable\_relationship.

#### **4.2.262.2 related**

The related specifies the second of the two Process\_variable (see 4.2.260) objects related by the Process\_variable\_relationship.

See 4.3.1813 for the application assertion.

#### **4.2.262.3 relating**

The relating specifies the first of the two Process\_variable (see 4.2.260) objects related by the Process\_variable\_relationship.

See 4.3.1814 for the application assertion.

#### **4.2.262.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- decomposition;
- substitution.

NOTE 1 See 4.2.262.4.1 - 4.2.262.4.3 for the definition of each predefined value for relation\_type.

##### **4.2.262.4.1 alternate**

alternate: The Process\_variable\_relationship defines a relationship where the related Process\_variable (see 4.2.260) is a possible substitute to the relating Process\_variable (see 4.2.260).

NOTE 2 This concept refers to the possibility to replace the Process\_variable (see 4.2.260). The actual replacement is addressed by 'substitution'.

##### **4.2.262.4.2 decomposition**

decomposition: The Process\_variable\_relationship defines a relationship where the related Process\_variable (see 4.2.260) is one of the components into which the relating Process\_variable (see 4.2.260) is divided.

##### **4.2.262.4.3 substitution**

substitution: The Process\_variable\_relationship defines a relationship where the related Process\_variable (see 4.2.260) replaces the relating Process\_variable (see 4.2.260).

### 4.2.263 Process\_variable\_system\_assignment

A Process\_variable\_system\_assignment is an association of a Process\_variable (see 4.2.260) with an item, such that the item generates, transforms, transfers, or consumes the Process\_variable (see 4.2.260).

The data associated with a Process\_variable\_system\_assignment are the following:

- associated\_process\_variable;
- associated\_system;
- description;
- role.

#### 4.2.263.1 associated\_process\_variable

The associated\_process\_variable specifies the Process\_variable (see 4.2.260).

See 4.3.1823 for the application assertion.

#### 4.2.263.2 associated\_system

The associated\_system specifies the item that processes or generates the Process\_variable (see 4.2.260).

Each associated\_system may be one of the following: Connectivity\_definition (see 4.2.61), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Function\_definition (see 4.2.145), Function\_unit (see 4.2.148), Functional\_connectivity\_definition (see 4.2.152), Physical\_instance (see 4.2.243), Port (see 4.2.247), Technical\_system (see 4.2.336), or Terminal (see 4.2.338).

See 4.3.1815, 4.3.1816, 4.3.1817, 4.3.1818, 4.3.1819, 4.3.1820, 4.3.1821, 4.3.1822, 4.3.1824, and 4.3.1825 for the application assertions.

#### 4.2.263.3 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Process\_variable\_system\_assignment.

The description need not be specified for a particular Process\_variable\_system\_assignment.

#### 4.2.263.4 role

The role specifies the relationship between the Process\_variable (see 4.2.260) and the associated system. The value is either user defined or predefined.



The predefined value of role is one of the following:

- amplifier;
- monitor;
- source;
- target;
- transmitter.

NOTE See 4.2.263.4.1 - 4.2.263.4.5 for the definition of each predefined value for role.

#### **4.2.263.4.1 amplifier**

amplifier: The process variable is boosted within the assigned system.

#### **4.2.263.4.2 monitor**

monitor: The process variable is observed within the assigned system.

#### **4.2.263.4.3 source**

source: The assigned system acts as a source for the process variable.

EXAMPLE A sensor can be a source for a process variable.

#### **4.2.263.4.4 target**

target: The assigned system acts as a target for the process variable.

EXAMPLE A screen that visualizes a message can be a target for a process variable.

#### **4.2.263.4.5 transmitter**

transmitter: The process variable is transferred within the assigned system.

### **4.2.264 Product\_class**

A Product\_class is the identification of a set of similar products to be offered to the market.

## ISO 10303-212:2001(E)

NOTE `Product_class` is the application object to which the definitions required for configuration control pertain.

The data associated with a `Product_class` are the following:

- description;
- id;
- level\_type;
- name;
- version\_id.

### 4.2.264.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the `Product_class`.

The description need not be specified for a particular `Product_class`.

### 4.2.264.2 id

The id specifies the identifier of the `Product_class`.

### 4.2.264.3 level\_type

The level\_type specifies the level or category of this `Product_class` in a hierarchical structure of `Product_class` objects.

EXAMPLE A high-level `Product_class` (level\_type = 'enterprise') may be used to define all `Specification_category` (see 4.2.324) and `Specification` (see 4.2.323) objects for all `Product_class` objects of an enterprise with several brands and companies.

A second-level `Product_class` (level\_type = 'platform') may be used to group all products that are based on the same technical concept (platform); these products may belong to different brands.

A third level `Product_class` (level\_type = 'product family') may be used to group all products that have a common base and a fixed set of characteristics (specification categories).

A fourth-level `Product_class` (level\_type = 'switch type') may represent products that are offered to the market; this level of `Product_class` may be defined by the marketing department. A set of specifications sufficient to produce the car is associated with its `Product_class`. Within this association, a distinction is made between standard characteristics (e.g., housing) and options that have to be chosen (e.g., colour) or that may be chosen by the customer (e.g., number of contacts).

### 4.2.264.4 name

The name specifies a speaking designation of the `Product_class`.

The name need not be specified for a particular Product\_class.

#### **4.2.264.5 version\_id**

The version\_id specifies versioning information for the Product\_class.

The version\_id need not be specified for a particular Product\_class.

#### **4.2.265 Product\_class\_relationship**

A Product\_class\_relationship is the relation between two Product\_class (see 4.2.263) objects.

The data associated with a Product\_class\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

##### **4.2.265.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Product\_class\_relationship.

The description need not be specified for a particular Product\_class\_relationship.

##### **4.2.265.2 related**

The related specifies the second of the two Product\_class (see 4.2.263) objects related by the Product\_class\_relationship.

See 4.3.1826 for the application assertion.

##### **4.2.265.3 relating**

The relating specifies the first of the two Product\_class (see 4.2.263) objects related by the Product\_class\_relationship.

See 4.3.1827 for the application assertion.

##### **4.2.265.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- decomposition;
- derivation;
- hierarchy;
- substitution;
- version sequence.

NOTE 1 See 4.2.265.4.1 - 4.2.265.4.5 for the definition of each predefined value for `relation_type`.

### 4.2.265.4.1 decomposition

decomposition: The `Product_class_relationship` defines a relationship where the related `Product_class` (see 4.2.263) is one of the possible variations of the relating `Product_class` (see 4.2.263).

### 4.2.265.4.2 derivation

derivation: The `Product_class_relationship` defines a deriving relationship where the related `Product_class` (see 4.2.263) is based on the relating `Product_class` (see 4.2.263).

### 4.2.265.4.3 hierarchy

hierarchy: The `Product_class_relationship` defines a relationship where the relating `Product_class` (see 4.2.263) is on a higher level in the hierarchy of `Product_class` (see 4.2.263) objects than the related `Product_class` (see 4.2.263).

### 4.2.265.4.4 substitution

substitution: The `Product_class_relationship` defines a relationship where the related `Product_class` (see 4.2.263) replaces the relating `Product_class` (see 4.2.263).

### 4.2.265.4.5 version sequence

version sequence: The `Product_class_relationship` defines a relationship where the relating `Product_class` (see 4.2.263) is the preceding version and the related `Product_class` (see 4.2.263) is the following version.

NOTE 2 The relationship does not imply inheritance of any kind between the application objects that are related.

## 4.2.266 Product\_component

A `Product_component` is a type of `Complex_product` (see 4.2.51) that is an element in the product decomposition structure. A `Product_component` may be represented by a set of `Alternative_solution` (see 4.2.12) objects with common functional requirements. The top-level `Product_component` of the decomposition tree shall be associated to a `Product_class` (see 4.2.263) as root entry. The

corresponding decomposition structure is identical for all variations of all products of that Product\_class (see 4.2.263).

The data associated with a Product\_component are the following:

- description;
- extended\_designation;
- instance\_required;
- is\_influenced\_by;
- is\_relevant\_for;
- name.

#### 4.2.266.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Product\_component.

The description need not be specified for a particular Product\_component.

#### 4.2.266.2 extended\_designation

The extended\_designation specifies a structured label for the Product\_component.

NOTE 1 The label assigned through extended\_designation shall be identical to the label assigned by the 'name' attribute.

The extended\_designation need not be specified for a particular Product\_component.

See 4.3.1830 for the application assertion.

#### 4.2.266.3 instance\_required

The instance\_required specifies whether or not the existence of a corresponding functional or physical item is required for the various Alternative\_solution (see 4.2.12) objects that provide implementations of the Product\_component.

NOTE 2 Product\_component objects that are not realized by parts may exist. In such cases, no instance of an item is required.

EXAMPLE Holes for wire harnesses are examples of Product\_component objects that do not require instances.

#### 4.2.266.4 is\_influenced\_by

The is\_influenced\_by specifies the Specification\_category (see 4.2.324) objects that impact the design of a solution for the Product\_component in the context of the Product\_class (see 4.2.263) objects that are referred to by the Class\_category\_association (see 4.2.40). These Product\_class (see

## ISO 10303-212:2001(E)

4.2.263) objects shall belong to the set of Product\_class (see 4.2.263) objects the Product\_component is valid for or shall be subclasses of these Product\_class (see 4.2.263) objects.

NOTE 3 All solutions for the Product\_component (i.e. the design of these solutions) are influenced by the referenced Specification\_category (see 4.2.324).

See 4.3.1829 for the application assertion.

### 4.2.266.5 is\_relevant\_for

The is\_relevant\_for specifies the Application\_context (see 4.2.22) in which the Product\_component must be considered.

See 4.3.1828 for the application assertion.

### 4.2.266.6 name

The name specifies a designation of the Product\_component.

The name need not be specified for a particular Product\_component.

### 4.2.267 Product\_constituent

A Product\_constituent is an object that may participate in the functional, logical, or physical breakdown or be an alternate realization of a Complex\_product (see 4.2.51).

Each Product\_constituent is either a Device (see 4.2.88), a Function\_unit (see 4.2.148), a Physical\_instance (see 4.2.243), or a Product\_component (see 4.2.265).

### 4.2.268 Product\_design

A Product\_design is the mechanism to associate an Item\_version (see 4.2.182) with its corresponding Product\_identification (see 4.2.268), where the specification is met by the design item.

The data associated with a Product\_design are the following:

— design;

— product.

#### 4.2.268.1 design

The design specifies the Item\_version (see 4.2.182) that meets the requirements.

See 4.3.1831 for the application assertion.

#### 4.2.268.2 product

The product specifies the Product\_identification (see 4.2.268) that represents the requirements.

See 4.3.1832 for the application assertion.

### 4.2.269 Product\_identification

A Product\_identification identifies an item that is manufacturable or expected as being manufacturable. A Product\_identification is defined with respect to the Product\_class (see 4.2.263) of which it is a member.

NOTE 1 The type of product that is to be manufactured from the data of the Product\_identification is the type of product which is identified through the 'associated\_product\_class' attribute.

NOTE 2 The intent of Product\_identification is the identification of a manufacturable item whereas the intent of Product\_class (see 4.2.263) is the gathering of products with similar characteristics.

EXAMPLE A Product\_identification associated with a Product\_class (see 4.2.263) identifying a family of asynchronous motors is intended to lead to the manufacture of a specific type of asynchronous motor.

NOTE 3 Let 'actual\_specification' be the set of instances obtained as the union of the Specification (see 4.2.323) objects related to the 'associated\_product\_class' through a Class\_specification\_association (see 4.2.44) or a Class\_condition\_association (see 4.2.41) with association type 'identification'.

The specification of a Product\_identification may be incomplete: there may not be in 'actual\_specification' any instance of Specification (see 4.2.323) for some Specification\_category (see 4.2.324) declared as mandatory for the 'associated\_product\_class'. For these Specification\_category (see 4.2.324) objects any Specification (see 4.2.323) available in the context of the 'associated product class' is considered as valid for the manufacturing of the product characterized by the considered Product\_identification.

The data associated with a Product\_identification are the following:

- associated\_product\_class;
- description;
- id;
- name;
- version\_id.

#### 4.2.269.1 associated\_product\_class

The associated\_product\_class specifies the Product\_class (see 4.2.263) that a product belongs to.

See 4.3.1833 for the application assertion.

#### 4.2.269.2 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Product\_identification.

## ISO 10303-212:2001(E)

The description need not be specified for a particular Product\_identification.

### 4.2.269.3 id

The id specifies the identifier of the Product\_identification.

### 4.2.269.4 name

The name specifies a speaking designation of the Product\_identification.

The name need not be specified for a particular Product\_identification.

### 4.2.269.5 version\_id

The version\_id specifies the identification of a particular version of a Product\_identification.

The version\_id need not be specified for a particular Product\_identification.

## 4.2.270 Product\_specification

A Product\_specification is a type of Product\_identification (see 4.2.268) that is the characterization of a manufacturable product for which one or more additional Specification (see 4.2.323) objects enhance the characterization provided for the associated Product\_class (see 4.2.263).

NOTE Let 'actual\_specification' be the set of instances of Specification (see 4.2.323) obtained as the union of 'defining\_specifications' and of the Specification (see 4.2.323) objects related to the 'associated\_product\_class' through a Class\_specification\_association (see 4.2.44) or a Class\_condition\_association (see 4.2.41) with association type 'identification'.

The specification of a Product\_specification may be incomplete: there may not be in 'actual\_specification' any instance of Specification (see 4.2.323) for some Specification\_category (see 4.2.324) declared as mandatory for the 'associated\_product\_class'. For these Specification\_category (see 4.2.324) objects any Specification (see 4.2.323) available in the context of the 'associated\_product\_class' is considered to be valid for the manufacturing of the product characterized by the considered Product\_specification.

EXAMPLE If the colour is not specified for a given family of switches, the equipment resulting from the realization of a Product\_specification associated to the Product\_class (see 4.2.263) representing the family of switches may be without paint or painted in any available colour.

The data associated with a Product\_specification are the following:

— defining\_specification.

### 4.2.270.1 defining\_specification

The defining\_specification specifies the set of Specification (see 4.2.323) objects necessary to discriminate the Product\_specification within its Product\_class (see 4.2.263).

See 4.3.1834 for the application assertion.



## 4.2.271 Product\_structure\_relationship

A Product\_structure\_relationship is an association between a Complex\_product (see 4.2.51) and a Product\_constituent (see 4.2.266), in which the Product\_constituent (see 4.2.266) is a functional, logical, or physical component or a realization of the Complex\_product (see 4.2.51).

The data associated with an Product\_structure\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.271.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Product\_structure\_relationship.

The description need not be specified for a particular Product\_structure\_relationship.

### 4.2.271.2 related

The related specifies the Product\_constituent (see 4.2.266) that is a functional, logical, or physical component or a realization of the relating Complex\_product (see 4.2.51).

NOTE - The semantics of this attribute are defined by the attribute relation\_type.

See 4.3.1836 for the application assertion.

### 4.2.271.3 relating

The relating specifies the Complex\_product (see 4.2.51) that is decomposed functionally, logically, or physically into or realized by the related Product\_constituent (see 4.2.266).

NOTE - The semantics of this attribute are defined by the attribute relation\_type.

See 4.3.1835 for the application assertion.

### 4.2.271.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of relation\_type is one of the following:

- decomposition;
- realization;
- specialization.

NOTE See 4.2.271.4.1 - 4.2.271.4.3 for the definition of each predefined value for relation\_type.

### 4.2.271.4.1 decomposition

decomposition: The related Product\_constituent (see 4.2.266) is one of potentially more components of the relating Complex\_product (see 4.2.51). This relation type shall only be used for Complex\_product (see 4.2.51) and Product\_constituent (see 4.2.266) of the same type.

### 4.2.271.4.2 realization

realization: The related Product\_constituent (see 4.2.266) is a means for fulfilling, either partially or fully, the requirements identified with the relating Complex\_product (see 4.2.51). This relation type shall be used only when the Complex\_product (see 4.2.51) and the Product\_constituent (see 4.2.266) are of different types.

### 4.2.271.4.3 specialization

specialization: The related Product\_constituent (see 4.2.266) fulfils the requirements of the relating Complex\_product (see 4.2.51) in a more specific way than defined for the relating Complex\_product (see 4.2.51). This relation type shall only be used for Product\_constituent (see 4.2.266) and Complex\_product (see 4.2.51) of the same type.

## 4.2.272 Project

A Project is a unique process with a time limit, a defined goal, a defined budget, and defined resources.

EXAMPLE For the development of a new product, a project is set up that is responsible for the development decisions as well as for the accounting of the costs.

The data associated with a Project are the following:

- actual\_end\_date;
- actual\_start\_date;
- affected\_product\_class;
- description;
- id;
- name;
- planned\_end\_date;
- planned\_start\_date;
- work\_program.

#### **4.2.272.1 actual\_end\_date**

The actual\_end\_date specifies the date when the Project was actually finished.

The actual\_end\_date need not be specified for a particular Project.

See 4.3.1838 for the application assertion.

#### **4.2.272.2 actual\_start\_date**

The actual\_start\_date specifies the date when the Project was actually started.

The actual\_start\_date need not be specified for a particular Project.

See 4.3.1839 for the application assertion.

#### **4.2.272.3 affected\_product\_class**

The affected\_product\_class specifies the Product\_class (see 4.2.263) that is affected by the work carried out within the Project.

See 4.3.1845 for the application assertion.

#### **4.2.272.4 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Project.

The description need not be specified for a particular Project.

#### **4.2.272.5 id**

The id specifies the identifier of the Project.

#### **4.2.272.6 name**

The name specifies a speaking designation of the Project.

#### **4.2.272.7 planned\_end\_date**

The planned\_end\_date specifies the date when the Project is or was supposed to be finished.

The planned\_end\_date need not be specified for a particular Project.

Each planned\_end\_date may be one of the following: Date\_time (see 4.2.79), Duration (see 4.2.126), or Event\_reference (see 4.2.130).

See 4.3.1840, 4.3.1842, and 4.3.1843 for the application assertions.

#### **4.2.272.8 planned\_start\_date**

The planned\_start\_date specifies the date when the Project is or was supposed to be started.

The planned\_start\_date need not be specified for a particular Project.

Each planned\_start\_date may be one of the following: Date\_time (see 4.2.79) or Event\_reference (see 4.2.130).

See 4.3.1841 and 4.3.1844 for the application assertions.

#### **4.2.272.9 work\_program**

The work program specifies the Activity (see 4.2.1) objects that are carried out within the Project.

See 4.3.1837 for the application assertion.

#### **4.2.273 Project\_relationship**

A Project\_relationship is the relation between two Project (see 4.2.271) objects.

EXAMPLE A team belongs to a department, which itself belongs to a company. This organizational structure can be built using the Organization\_relationship (see 4.2.225) mechanism.

The data associated with a Project\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### 4.2.273.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Project\_relationship.

The description need not be specified for a particular Project\_relationship.

#### 4.2.273.1 related

The related specifies the second of the two Project (see 4.2.271) objects related by a Project\_relationship.

NOTE The semantic of this attribute is defined by the relation\_type attribute.

See 4.3.1846 for the application assertion.

#### 4.2.273.2 relating

The relating specifies the first of the two Project (see 4.2.271) objects related by a Project\_relationship.

NOTE The semantic of this attribute is defined by the relation\_type attribute.

See 4.3.1847 for the application assertion.

#### 4.2.273.3 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- decomposition;
- dependency;
- sequence;
- succession.

NOTE See 4.2.273.3.1 - 4.2.273.3.4 for the definition of each predefined value for `relation_type`.

### 4.2.273.3.1 decomposition

decomposition: The `Project_relationship` defines a relationship where the related Project (see 4.2.271) is one of the components into which the relating Project (see 4.2.271) is broken down.

### 4.2.273.3.2 dependency

dependency: The `Project_relationship` defines a relationship where the related Project (see 4.2.271) is dependent of the relating Project (see 4.2.271).

### 4.2.273.3.3 sequence

sequence: The `Project_relationship` defines a relationship where the relating Project (see 4.2.271) shall be completed before the related Project (see 4.2.271) starts.

### 4.2.273.3.4 succession

succession: The `Project_relationship` defines a relationship where the related Project (see 4.2.271) is the successor of the relating Project (see 4.2.271).

## 4.2.274 Projection\_line

A `Projection_line` is a type of `Directed_curve` (see 4.2.99) that represents the extension of a point, line, surface, or theoretical point of intersection to a location outside the part outline.

The data associated with a `Projection_line` are the following:

- `projected_element`.

### 4.2.274.1 projected\_element

The `projected_element` specifies the geometric or annotation element that is projected by the projection line.

The `projected_element` need not be specified for a particular `Projection_line`.

See 4.3.1848 for the application assertion.

### 4.2.275 Promissory\_usage

A Promissory\_usage is a type of Assembly\_component\_relationship (see 4.2.26) that is the relation between a constituent of an assembly and the assembly itself. The relationship describes the intention to use the constituent in an assembly. No geometric information is required for the constituent or the assembly that is associated by a Promissory\_usage.

EXAMPLE In early design stages the Promissory\_usage can be used to create preliminary bills of material for prototyping purpose.

### 4.2.276 Property\_reference

The Property\_reference specifies the information that is required to retrieve a Data\_element - definition (see 4.2.72) object from a library compliant to ISO 13584-42.

NOTE 1 The library is not necessarily available in computer interpretable form.

The data associated with a Property\_reference are the following:

- code;
- name\_scope;
- version.

#### 4.2.276.1 code

The code specifies a computer-interpretable identifier for the General\_classification (see 4.2.156) object within the repository. The format of this code is defined in ISO 13584-42.

NOTE 2 The type of repository referred within the associated Class\_reference (see 4.2.43) may impose further restrictions on the content of this attribute.

#### 4.2.276.1 name\_scope

The name\_scope specifies the Class\_reference (see 4.2.43) in which the property is visible.

See 4.3.1849 for the application assertion.

#### 4.2.276.1 version

The version specifies the variant of the entry in the repository. The format of this version is defined in ISO 13584-42.

NOTE 3 The type of repository referred within the associated Class\_reference (see 4.2.43) may impose further restrictions on the content of this attribute.

### 4.2.277 Quantified\_device

A Quantified\_device is a type of Device (see 4.2.88) that allows the aggregation of equipment.

## ISO 10303-212:2001(E)

EXAMPLE Rivets are not normally addressed individually but as a quantified device.

The data associated with a Quantified\_device are the following:

— quantity.

### 4.2.277.1 quantity

The quantity specifies the value and the unit that stipulate the amount of occurrences.

See 4.3.1850 for the application assertion.

### 4.2.278 Radius\_dimension

A Radius\_dimension is a type of Dimension (see 4.2.93) that is the graphical presentation of the value of the radial distance from the centre of a circular element to a point on the element.

The data associated with a Radius\_dimension are the following:

— component;

— extent.

#### 4.2.278.1 component

The component specifies the projection lines that shows the extension of the points, lines, or surfaces that bound the measurement.

The component need not be specified for a particular Radius\_dimension.

See 4.3.1852 for the application assertion.

#### 4.2.278.2 extent

The extent specifies the dimension line that graphically presents where the dimension value applies.

See 4.3.1851 for the application assertion.

### 4.2.279 Rated\_current

A Rated\_current is the magnitude of the electrical current establishing a basis for the design of a piece of equipment.

The data associated with a Rated\_current are the following:

— type\_of Rated\_current;

— value\_of Rated\_current.

#### 4.2.279.1 type\_of Rated\_current

The type\_of Rated\_current specifies the kind of electrical current specified by Rated\_current.



The value of `type_of Rated_current` is one of the following:

- ac 3 phase;
- ac 5 phases;
- ac single phase;
- dc.

NOTE See 4.2.279.1.1 - 4.2.279.1.4 for the definition of each permissible value for `type_of Rated_current`.

#### **4.2.279.1.1 ac 3 phase**

ac 3 phase: The `Rated_current` is a three-phase alternating current.

#### **4.2.279.1.2 ac 5 phases**

ac 5 phases: The `Rated_current` is a five-phase alternating current.

#### **4.2.279.1.3 ac single phase**

ac single phase: The `Rated_current` is a single-phase alternating current.

#### **4.2.279.1.4 dc**

dc: The `Rated_current` is a direct current.

### **4.2.279.2 value\_of Rated\_current**

The `value_of Rated_current` specifies the magnitude of the electrical current.

See 4.3.1853 for the application assertion.

### **4.2.280 Rated\_power**

A `Rated_power` is the magnitude of the electrical power establishing a basis for the design of a piece of electrotechnical equipment.

The data associated with a `Rated_power` are the following:

- `type_of Rated_power`;
- `value_of Rated_power`.

#### **4.2.280.1 type\_of Rated\_power**

The `type_of Rated_power` specifies the kind of electrical power specified by `Rated_power`.

## ISO 10303-212:2001(E)

The value of `type_of Rated_power` is one of the following:

- apparent power;
- reactive power;
- true power.

NOTE See 4.2.280.1.1 - 4.2.280.1.3 for the definition of each permissible value for `type_of Rated_power`.

### 4.2.280.1.1 apparent power

apparent power: The `Rated_power` is the electrical power that is the product of the r.m.s values of voltage and current.

### 4.2.280.1.2 reactive power

reactive power: The `Rated_power` is the imaginary part of the complex power.

### 4.2.280.1.3 true power

true power: The `Rated_power` is the mean value, taken over one period, of the instantaneous power.

## 4.2.280.2 value\_of Rated\_power

The `value_of Rated_power` specifies the magnitude of the electrical power.

See 4.3.1854 for the application assertion.

## 4.2.281 Rated\_voltage

A `Rated_voltage` is the magnitude of the electrical voltage establishing a basis for the design of an electrical device.

The data associated with a `Rated_voltage` are the following:

- `type_of Rated_voltage`;
- `value_of Rated_voltage`.

### 4.2.281.1 type\_of Rated\_voltage

The `type_of Rated_voltage` specifies the kind of electrical voltage specified by `Rated_voltage`.

The value of `type_of Rated_voltage` is one of the following:

- ac 3 phase;
- ac 5 phase;
- ac single phase;
- dc.

NOTE See 4.2.281.1.1 - 4.2.281.1.5 for the definition of each permissible value for `type_of Rated_voltage`.

#### **4.2.281.1.1 ac 3 phase**

ac 3 phase: The `Rated_voltage` is a three-phase alternating voltage.

#### **4.2.281.1.2 ac 5 phase**

ac 5 phase: The `Rated_voltage` is a five-phase alternating voltage.

#### **4.2.281.1.3 ac single phase**

ac single phase: The `Rated_voltage` is a single-phase alternating voltage.

#### **4.2.281.1.4 dc**

dc: The `Rated_voltage` is a direct voltage.

### **4.2.281.2 value\_of Rated\_voltage**

The `value_of Rated_voltage` specifies the magnitude of the electrical voltage.

See 4.3.1855 for the application assertion.

### **4.2.282 Rectangular\_area**

A `Rectangular_area` is a plane area with four straight sides and four right angles, especially one with unequal adjacent sides.

The data associated with a `Rectangular_area` are the following:

- height;
- position;
- width.

#### **4.2.282.1 height**

The height specifies the tallness of the area.

#### **4.2.282.2 position**

The position specifies the placement of the Rectangular\_area.

#### **4.2.282.3 width**

The width specifies the broadness of the area.

#### **4.2.283 Rectangular\_size**

A Rectangular\_size is the extent of a rectangular area.

The data associated with a Rectangular\_size are the following:

- density;
- height;
- width.

##### **4.2.283.1 density**

The density specifies the resolution of the object if it is a raster picture.

The density need not be specified for a particular Rectangular\_size.

##### **4.2.283.2 height**

The height specifies the size of the object in vertical direction.

##### **4.2.283.3 width**

The width specifies the size of the object in horizontal direction.

#### **4.2.284 Reference\_grid**

A Reference\_grid is rectangular pattern of lines that provides a grid reference system in accordance with IEC 61082-1. The grid reference system allows one to identify a location on a diagram.

The data associated with a Reference\_grid are the following:

- column\_id;
- grid\_origin;
- row\_id.

##### **4.2.284.1 column\_id**

The column\_id specifies the identifier for each column of the reference grid. The identifiers are contained in a list representing the order of columns from left to right of the sheet.

There shall be one or more `column_id` for a `Reference_grid`.

#### **4.2.284.2 `grid_origin`**

The `grid_origin` specifies a fixed point from which coordinates representing the `Reference_grid` are measured.

#### **4.2.284.3 `row_id`**

The `row_id` specifies the identifier for each row of the reference grid. The identifiers are contained in a list representing the order of rows from bottom to the top of the sheet.

There shall be one or more `row_id` for a `Reference_grid`.

#### **4.2.285 `Reference_grid_layout`**

A `Reference_grid_layout` is a type of `User_defined_symbol_definition` (see 4.2.355) that depicts the arrangement and design of annotation elements that visualize the `Reference_grid` (see 4.2.283) on a diagram.

The data associated with a `Reference_grid_layout` are the following:

— `assigned_reference_grid`.

##### **4.2.285.1 `assigned_reference_grid`**

The `assigned_reference_grid` specifies the `Reference_grid` (see 4.2.283) object of which the `reference_grid_layout` is the pictorial representation.

See 4.3.1856 for the application assertion.

#### **4.2.286 `Requirement`**

A `Requirement` is human-interpretable product data that describes constraints that the electrotechnical system or some elements of the system may satisfy.

**NOTE** By making use of `Requirement` objects the design rationale of a system as a whole or of portions thereof can be laid down.

## ISO 10303-212:2001(E)

EXAMPLE An error handling system shall provide for a buffer capable to handle 2000 error messages per second for a period of 48 hours.

The data associated with a Requirement are the following:

- description;
- id;
- name;
- version\_id.

### 4.2.286.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Requirement.

The description need not be specified for a particular Requirement.

### 4.2.286.2 id

The id specifies the identifier of the Requirement.

### 4.2.286.3 name

The name specifies a speaking designation of the Requirement.

The name need not be specified for a particular Requirement.

### 4.2.286.4 version\_id

The version\_id specifies versioning information for the Requirement.

The version\_id need not be specified for a particular Requirement.

## 4.2.287 Requirement\_assignment

A Requirement\_assignment is a relation that associates a Requirement (see 4.2.285) with an physical or abstract item.

The data associated with a Requirement\_assignment are the following:

- associated\_requirement;
- constrained\_element;
- description;
- relation\_type.

### 4.2.287.1 associated\_requirement

The associated\_requirement specifies the Requirement (see 4.2.285).

See 4.3.1902 for the application assertion.

### 4.2.287.2 constrained\_element

The constrained\_element specifies the element to which the Requirement (see 4.2.285) applies.

Each constrained\_element may be one of the following: Assembly\_component\_relationship (see 4.2.26), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Data\_element (see 4.2.70), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_version (see 4.2.114), Drawing\_sheet (see 4.2.122), Drawing\_view (see 4.2.125), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Functionality (see 4.2.155), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Path (see 4.2.232), Path\_node (see 4.2.233), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Process\_variable (see 4.2.260), Product\_class (see 4.2.263), Product\_specification (see 4.2.269), Product\_structure\_relationship (see 4.2.270), Route (see 4.2.290), Route\_relationship (see 4.2.291), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Signal (see 4.2.309), Signal\_value (see 4.2.313), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), or Terminal (see 4.2.338).

See 4.3.1857, 4.3.1858, 4.3.1859, 4.3.1860, 4.3.1861, 4.3.1862, 4.3.1863, 4.3.1864, 4.3.1865, 4.3.1866, 4.3.1867, 4.3.1868, 4.3.1869, 4.3.1870, 4.3.1871, 4.3.1872, 4.3.1873, 4.3.1874, 4.3.1875, 4.3.1876, 4.3.1877, 4.3.1878, 4.3.1879, 4.3.1880, 4.3.1881, 4.3.1882, 4.3.1883, 4.3.1884, 4.3.1885, 4.3.1886, 4.3.1887, 4.3.1888, 4.3.1889, 4.3.1890, 4.3.1891, 4.3.1892, 4.3.1893, 4.3.1894, 4.3.1895, 4.3.1896, 4.3.1897, 4.3.1898, 4.3.1899, 4.3.1900, 4.3.1901, 4.3.1903, 4.3.1904, 4.3.1905, 4.3.1906, 4.3.1907, 4.3.1908, 4.3.1909, 4.3.1910, 4.3.1911, 4.3.1912, 4.3.1913, and 4.3.1914 for the application assertions.

### 4.2.287.3 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Requirement\_assignment.

The description need not be specified for a particular Requirement\_assignment.

### 4.2.287.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- compelled element;
- solution.

NOTE See 4.2.287.4.1 - 4.2.287.4.2 for the definition of each predefined value for `relation_type`.

### 4.2.287.4.1 compelled element

compelled element: The `Requirement_assignment` defines a relationship where the associated Requirement (see 4.2.285) levies a constraint against the associated item.

### 4.2.287.4.2 solution

solution: The `Requirement_assignment` defines a relationship where the `constrained_element` serves as a means to solve the associated Requirement (see 4.2.285).

## 4.2.288 Requirement\_document\_assignment

A `Requirement_document_assignment` is a relation that associates a Requirement (see 4.2.285) with a document.

The data associated with a `Requirement_document_assignment` are the following:

- description;
- documentation;
- documented\_requirement;
- role\_of\_document.

### 4.2.288.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the `Requirement_document_assignment`.

The description need not be specified for a particular `Requirement_document_assignment`.

### 4.2.288.2 documentation

The documentation specifies the record that deals with the Requirement (see 4.2.285).

Each documentation may be one of the following: Document (see 4.2.101), Document\_file (see 4.2.106), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), or Drawing (see 4.2.119).

See 4.3.1915, 4.3.1916, 4.3.1917, 4.3.1918, and 4.3.1919 for the application assertions.



### 4.2.288.3 documented\_requirement

The documented\_requirement specifies the Requirement (see 4.2.285) object the document deals with.

See 4.3.1920 for the application assertion.

### 4.2.288.4 role\_of\_document

The role\_of\_document specifies the function of the document in respect of the Requirement (see 4.2.285). The value is either user defined or predefined.

The role\_of\_document need not be specified for a particular Requirement\_document\_assignment. The role\_of\_document need not be specified for a particular Requirement\_document\_assignment.

The predefined value of role\_of\_document is one of the following:

- additional information;
- specification;
- verification.

NOTE See 4.2.288.4.1 - 4.2.288.4.3 for the definition of each predefined value for role\_of\_document.

#### 4.2.288.4.1 additional information

additional information: The document provides further information on the Requirement (see 4.2.285).

#### 4.2.288.4.2 specification

specification: The document raises the Requirement (see 4.2.285).

#### 4.2.288.4.3 verification

verification: The document contains information that proves the fulfilment of the Requirement (see 4.2.285).

### 4.2.289 Requirement\_relationship

A Requirement\_relationship is the relation between two Requirement (see 4.2.285) objects.

## ISO 10303-212:2001(E)

The data associated with an Requirement\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.289.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Requirement\_relationship.

The description need not be specified for a particular Requirement\_relationship.

### 4.2.289.2 related

The related specifies the second of the two Requirement (see 4.2.285) objects related by the Requirement\_relationship.

See 4.3.1921 for the application assertion.

### 4.2.289.3 relating

The relating specifies the first of the two Requirement (see 4.2.285) objects related by the Requirement\_relationship.

See 4.3.1922 for the application assertion.

### 4.2.289.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution;
- version hierarchy;
- version sequence.

NOTE 1 See 4.2.289.4.1 - 4.2.289.4.6 for the definition of each predefined value for relation\_type.

**4.2.289.4.1 alternate**

alternate: The Requirement\_relationship defines a relationship where the related Requirement (see 4.2.285) is a possible substitute to the relating Requirement (see 4.2.285).

NOTE 2 This concept refers to the possibility to replace the Requirement (see 4.2.285). The actual replacement is addressed by 'substitution'.

**4.2.289.4.2 decomposition**

decomposition: The Requirement\_relationship defines a relationship where the related Requirement (see 4.2.285) is one of the components into which the relating Requirement (see 4.2.285) is divided.

**4.2.289.4.3 derivation**

derivation: The Requirement\_relationship defines a deriving relationship where the related Requirement (see 4.2.285) is based on the relating Requirement (see 4.2.285).

**4.2.289.4.4 substitution**

substitution: The Requirement\_relationship defines a relationship where the related Requirement (see 4.2.285) replaces the relating Requirement (see 4.2.285).

**4.2.289.4.5 version hierarchy**

version hierarchy: The Requirement\_relationship defines a hierarchical relationship where the related Requirement (see 4.2.285) is a subversion of the relating Requirement (see 4.2.285).

EXAMPLE Revision 1.1 and 1.2 of a requirement.

**4.2.289.4.6 version sequence**

version sequence: The Requirement\_relationship defines a succession of versions where the relating Requirement (see 4.2.285) is the preceding version and the related Requirement (see 4.2.285) is the following version. For a Requirement (see 4.2.285), there shall be, at the most, one Requirement\_relationship of this relation type as relating and, at most, one Requirement\_relationship of this relation type as related.

**4.2.290 Retention\_period**

A Retention period is the definition of the period of time that product data needs to be maintained due to organizational policy or legal requirements.

## ISO 10303-212:2001(E)

The data associated with a `Retention_period` are the following:

- `earliest_end_definition`;
- `is_applied_to`;
- `latest_end_definition`;
- `retention_purpose`;
- `start_definition`.

### 4.2.290.1 `earliest_end_definition`

The `earliest_end_definition` specifies the earliest point in time from which on all items, the `Retention_period` is applied to, may be deleted. In this context deletion applies to all subordinate items that are not referenced by other items.

Each `earliest_end_definition` may be one of the following: `Date_time` (see 4.2.79), `Duration` (see 4.2.126), or `Event_reference` (see 4.2.130).

See 4.3.1954, 4.3.1970, and 4.3.1972 for the application assertions.

### 4.2.290.2 `is_applied_to`

The `is_applied_to` specifies the items that are controlled by the `Retention_period`.

NOTE The master document is the one for which earliest end definition and latest end definition are the same.

Each `is_applied_to` may be one of the following: `Activity` (see 4.2.1), `Activity_element` (see 4.2.2), `Activity_method_assignment` (see 4.2.4), `Activity_relationship` (see 4.2.5), `Alternate_item_relationship` (see 4.2.11), `Approval_status` (see 4.2.25), `Assembly_component_relationship` (see 4.2.26), `Assembly_substitute_relationship` (see 4.2.28), `Cable_pull_information` (see 4.2.33), `Certification` (see 4.2.38), `Class_category_association` (see 4.2.40), `Class_condition_association` (see 4.2.41), `Class_inclusion_association` (see 4.2.42), `Class_specification_association` (see 4.2.44), `Class_structure_relationship` (see 4.2.45), `Classification_association` (see 4.2.46), `Classification_attribute` (see 4.2.47), `Classification_system` (see 4.2.48), `Complex_product` (see 4.2.51), `Complex_product_relationship` (see 4.2.52), `Composition_relationship` (see 4.2.55), `Configuration` (see 4.2.56), `Connectivity_allocation` (see 4.2.60), `Connectivity_definition` (see 4.2.61), `Connectivity_definition_relationship` (see 4.2.62), `Contract` (see 4.2.63), `Data_element` (see 4.2.70), `Data_element_association` (see 4.2.71), `Data_element_definition` (see 4.2.72), `Data_element_relationship` (see 4.2.74), `Data_element_specification` (see 4.2.75), `Design_discipline_item_definition` (see 4.2.86), `Device` (see 4.2.88), `Device_relationship` (see 4.2.89), `Document` (see 4.2.101), `Document_file` (see 4.2.106), `Document_file_relationship` (see 4.2.107), `Document_representation` (see 4.2.110), `Document_version` (see 4.2.114), `Document_version_relationship` (see 4.2.115), `Drawing` (see 4.2.119), `Drawing_sequence` (see 4.2.121), `Drawing_sheet` (see 4.2.122), `Drawing_sheet_relationship` (see 4.2.124), `Free_segment` (see 4.2.144), `Function_definition` (see 4.2.145), `Function_definition_relationship` (see 4.2.146), `Function_interface` (see 4.2.147), `Function_unit` (see 4.2.148), `Function_unit_relationship` (see 4.2.149), `Function_version` (see 4.2.150), `Function_version_relationship` (see 4.2.151), `Functional_connectivity_definition` (see 4.2.152), `Functional_connectivity_definition_relationship` (see 4.2.153), `Functional_unit_allocation` (see 4.2.154), `Functionality` (see 4.2.155),

General\_classification (see 4.2.156), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Marking (see 4.2.199), Material (see 4.2.201), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Organization\_relationship (see 4.2.225), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Person\_in\_organization (see 4.2.238), Person\_in\_organization\_relationship (see 4.2.239), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Port\_association (see 4.2.249), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Requirement\_document\_assignment (see 4.2.287), Route (see 4.2.290), Route\_relationship (see 4.2.291), Routed\_segment (see 4.2.293), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Security\_classification (see 4.2.301), Security\_level (see 4.2.302), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.1923, 4.3.1924, 4.3.1925, 4.3.1926, 4.3.1927, 4.3.1928, 4.3.1929, 4.3.1930, 4.3.1931, 4.3.1932, 4.3.1933, 4.3.1934, 4.3.1935, 4.3.1936, 4.3.1937, 4.3.1938, 4.3.1939, 4.3.1940, 4.3.1941, 4.3.1942, 4.3.1943, 4.3.1944, 4.3.1945, 4.3.1946, 4.3.1947, 4.3.1948, 4.3.1949, 4.3.1950, 4.3.1951, 4.3.1952, 4.3.1953, 4.3.1957, 4.3.1958, 4.3.1959, 4.3.1960, 4.3.1961, 4.3.1962, 4.3.1963, 4.3.1964, 4.3.1965, 4.3.1966, 4.3.1967, 4.3.1968, 4.3.1969, 4.3.1975, 4.3.1976, 4.3.1977, 4.3.1978, 4.3.1979, 4.3.1980, 4.3.1981, 4.3.1982, 4.3.1983, 4.3.1984, 4.3.1985, 4.3.1986, 4.3.1987, 4.3.1988, 4.3.1989, 4.3.1990, 4.3.1991, 4.3.1992, 4.3.1993, 4.3.1994, 4.3.1995, 4.3.1996, 4.3.1997, 4.3.1998, 4.3.1999, 4.3.2000, 4.3.2001, 4.3.2002, 4.3.2003, 4.3.2004, 4.3.2005, 4.3.2006, 4.3.2007, 4.3.2008, 4.3.2009, 4.3.2010, 4.3.2011, 4.3.2012, 4.3.2013, 4.3.2014, 4.3.2015, 4.3.2016, 4.3.2017, 4.3.2018, 4.3.2019, 4.3.2020, 4.3.2021, 4.3.2022, 4.3.2023, 4.3.2024, 4.3.2025, 4.3.2026, 4.3.2027, 4.3.2028, 4.3.2029, 4.3.2030, 4.3.2031, 4.3.2032, 4.3.2033, 4.3.2034, 4.3.2035, 4.3.2036, 4.3.2037, 4.3.2038, 4.3.2039, 4.3.2040, 4.3.2041, 4.3.2042, 4.3.2043, 4.3.2044, 4.3.2045, 4.3.2046, 4.3.2047, and 4.3.2048 for the application assertions.

### 4.2.290.3 latest\_end\_definition

The latest\_end\_definition specifies the latest point in time at which all items, to which the Retention\_period applies, are deleted. In this context deletion applies to all subordinate items that are not used by other items.

Each latest\_end\_definition may be one of the following: Date\_time (see 4.2.79), Duration (see 4.2.126), or Event\_reference (see 4.2.130).

See 4.3.1955, 4.3.1971, and 4.3.1973 for the application assertions.

### 4.2.290.4 retention\_purpose

The retention\_purpose specifies the rationale behind the Retention\_period.

## ISO 10303-212:2001(E)

The retention\_purpose need not be specified for a particular Retention\_period.

### 4.2.290.5 start\_definition

The start\_definition specifies the point in time at which the Retention\_period starts.

Each start\_definition may be one of the following: Date\_time (see 4.2.79) or Event\_reference (see 4.2.130).

See 4.3.1956 and 4.3.1974 for the application assertions.

### 4.2.291 Route

A Route is a named course taken to get from a starting point to a destination.

NOTE 1 A Route is understood as a sequence of vertices and edges, where the edges are Section (see 4.2.296) objects and the vertices are Node (see 4.2.208) objects. Routed\_object (see 4.2.292) objects allow for the specification of the items that make use of the Route.

EXAMPLE 1 A cable rack provides routes for cables. Some its routes are occupied, i.e. a cable is laid on that route, others may remain empty for future use. To specify that a cable is routed partially or as a whole it is assigned to Routed\_object (see 4.2.292) objects which in turn are associated to the Route.

NOTE 2 A Route can be used to specify a path travelled by a person for delivering, selling, or collecting goods or services.

EXAMPLE 2 A path travelled by a person throughout the commissioning of a power plant, e.g. for inspection purposes.

The data associated with a Route are the following:

- course;
- description;
- encountered\_object;
- id;
- version\_id.

#### 4.2.291.1 course

The course specifies the direction taken.

Each course may be one of the following: Node (see 4.2.208), Section (see 4.2.296), or Section\_interface (see 4.2.298).

See 4.3.2054, 4.3.2057, and 4.3.2058 for the application assertions.

### 4.2.291.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Route.

The description need not be specified for a particular Route.

### 4.2.291.3 encountered\_object

The encountered\_object specifies an ordered list of physical or abstract items the route comes in touch.

Each encountered\_object may be one of the following: Connectivity\_definition (see 4.2.61), Device (see 4.2.88), Function\_unit (see 4.2.148), Functional\_connectivity\_definition (see 4.2.152), Location (see 4.2.192), Physical\_instance (see 4.2.243), Product\_component (see 4.2.265), or Signal (see 4.2.309).

See 4.3.2049, 4.3.2050, 4.3.2051, 4.3.2052, 4.3.2053, 4.3.2055, 4.3.2056, and 4.3.2059 for the application assertions.

### 4.2.291.4 id

The id specifies the identifier of the Route.

### 4.2.291.5 version\_id

The version\_id specifies versioning information for the Route.

The version\_id need not be specified for a particular Route.

## 4.2.292 Route\_relationship

A Route\_relationship is the relation between two Route (see 4.2.290) objects.

The data associated with a Route\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.292.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Route\_relationship.

The description need not be specified for a particular Route\_relationship.

#### **4.2.292.2 related**

The related specifies the second of the two Route (see 4.2.290) objects related by the Route\_ - relationship.

See 4.3.2060 for the application assertion.

#### **4.2.292.3 relating**

The relating specifies the first of the two Route (see 4.2.290) objects related by the Route\_ - relationship.

See 4.3.2061 for the application assertion.

#### **4.2.292.4 relation\_type**

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution.

NOTE 1 See 4.2.292.4.1 - 4.2.292.4.4 for the definition of each predefined value for relation\_ - type.

##### **4.2.292.4.1 alternate**

alternate: The Route\_relationship defines a relationship where the related Route (see 4.2.290) is an alternative to the relating Route (see 4.2.290).

NOTE 2 This concept refers to the possibility to replace the Route (see 4.2.290). The actual replacement is addressed by 'substitution'.

##### **4.2.292.4.2 decomposition**

decomposition: The Route\_relationship defines a relationship where the related Route (see 4.2.290) is one of the components into which the relating Route (see 4.2.290) is divided.

##### **4.2.292.4.3 derivation**

derivation: The Route\_relationship defines a deriving relationship where the related Route (see 4.2.290) is based on the relating Route (see 4.2.290).



**4.2.292.4.4 substitution**

substitution: The `Route_relationship` defines a relationship where the related `Route` (see 4.2.290) replaces the relating `Route` (see 4.2.290).

**4.2.293 Routed\_object**

The `Routed_object` is an physical or abstract item to which routing information is assigned.

The data associated with a `Routed_object` are the following:

- `arrangement`;
- `associated_object`.

**4.2.293.1 arrangement**

The `arrangement` specifies an ordered list of sections that specifies the route of the `Routed_object`.

Each `arrangement` may be one of the following: `Free_segment` (see 4.2.144) or `Routed_segment` (see 4.2.293).

See 4.3.2064 and 4.3.2069 for the application assertions.

**4.2.293.2 associated\_object**

The `associated_object` specifies the item to which the routing information applies.

Each `associated_object` may be one of the following: `Connectivity_definition` (see 4.2.61), `Device` (see 4.2.88), `Function_unit` (see 4.2.148), `Functional_connectivity_definition` (see 4.2.152), `Physical_instance` (see 4.2.243), or `Product_component` (see 4.2.265).

See 4.3.2062, 4.3.2063, 4.3.2065, 4.3.2066, 4.3.2067, and 4.3.2068 for the application assertions.

**4.2.294 Routed\_segment**

A `Routed_segment` is a portion of a `Routed_object` (see 4.2.292) that has a well-defined course.

The data associated with a `Routed_segment` are the following:

- `course`;
- `description`;
- `id`.

**4.2.294.1 course**

The `course` specifies the way that is followed by the `Routed_segment`.

Each `course` may be one of the following: `Node` (see 4.2.208), `Route` (see 4.2.290), `Section` (see 4.2.296), or `Section_interface` (see 4.2.298).

## ISO 10303-212:2001(E)

See 4.3.2070, 4.3.2071, 4.3.2072, and 4.3.2073 for the application assertions.

### 4.2.294.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Routed\_segment.

The description need not be specified for a particular Routed\_segment.

### 4.2.294.3 id

The id specifies the identifier of the Routed\_segment.

### 4.2.295 Schematic\_node

A Schematic\_node is a type of Annotation\_element (see 4.2.15) that describes that part of a pictorial presentation to which connecting lines or terminals of other symbols may be connected.

The data associated with a Schematic\_node are the following:

- assigned\_to;
- definition;
- position;
- rotation;
- scale.

#### 4.2.295.1 assigned\_to

The assigned\_to specifies the User\_defined\_symbol (see 4.2.354) to which the Schematic\_node is associated.

The assigned\_to need not be specified for a particular Schematic\_node.

See 4.3.2076 for the application assertion.

#### 4.2.295.2 definition

The definition specifies the Typical\_schematic\_node (see 4.2.347) object that served as a template for the Schematic\_node.

See 4.3.2075 for the application assertion.

#### 4.2.295.3 position

The position specifies the location of the Schematic\_node relative to the coordinate system of the associated User\_defined\_symbol (see 4.2.354).

See 4.3.2074 for the application assertion.

**4.2.295.4 rotation**

The rotation specifies the angle, measured counter-clockwise, between the horizontal axis of the coordinate system in which the Schematic\_node is defined and the horizontal axis of the coordinate system into which the Schematic\_node is being placed.

**4.2.295.5 scale**

The scale specifies the ratio between the size of the Schematic\_node as defined and the size of the Schematic\_node as presented.

**4.2.296 Schematic\_text**

A Schematic\_text is a type of Text (see 4.2.342) that is written information in schematic diagrams. It is based on predefined templates that define the layout of the text and may contain default text.

The data associated with a Schematic\_text are the following:

- assigned\_to;
- definition.

**4.2.296.1 assigned\_to**

The assigned\_to specifies the User\_defined\_symbol (see 4.2.354) to which the Schematic\_node (see 4.2.294) is associated.

The assigned\_to need not be specified for a particular Schematic\_text.

Each assigned\_to may be one of the following: Schematic\_node (see 4.2.294) or User\_defined\_symbol (see 4.2.354).

See 4.3.2077 and 4.3.2079 for the application assertions.

**4.2.296.2 definition**

The definition specifies the template from which the Schematic\_text is derived.

See 4.3.2078 for the application assertion.

**4.2.297 Section**

A Section describes a segment of a Route (see 4.2.290). A Section may be curved.

## ISO 10303-212:2001(E)

NOTE 1 A Route (see 4.2.290) is understood as a sequence of vertices and edges, in which the edges are Section objects.

NOTE 2 Sections and Node (see 4.2.208) objects can be used to describe the three-dimensional route of a cable.

The data associated with a Section are the following:

- bending\_radius;
- course;
- cross\_section;
- description;
- id;
- implemented\_by;
- length\_of\_section;
- space\_factor;
- version\_id;

### 4.2.297.1 bending\_radius

The bending\_radius specifies a quantity that characterizes the radius of the curves a piece of equipment laid in the Section needs to follow.

NOTE 1 If more than one Single\_value (see 4.2.316) object is assigned different aspects of the bending radius are specified.

NOTE 2 The value of the 'bending\_radius' attribute can be used to determine the required flexibility of a cable that is to be laid into the Section.

EXAMPLE To define a particular cabletray the typical and the minimum value of the bending radius are specified.

See 4.3.2083 for the application assertion.

### 4.2.297.2 course

The course specifies the path of the Section.

The course need not be specified for a particular Section.

See 4.3.2082 for the application assertion.

**4.2.297.3 cross\_sectional\_area**

The `cross_sectional_area` describes the transverse section of a Section in which the cabling can be placed.

NOTE If more than one `Cross_section` (see 4.2.65) object is assigned different aspects of the transverse section are specified.

EXAMPLE To define a particular cabletray the typical and the minimum value of the cross section are specified.

See 4.3.2080 for the application assertion.

**4.2.297.4 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Section.

The description need not be specified for a particular Section.

**4.2.297.5 id**

The `id` specifies the identifier of the Section.

**4.2.297.6 implemented\_by**

The `implemented_by` specifies the equipment that is used to realize the Section.

See 4.3.2081 for the application assertion.

**4.2.297.7 length\_of\_section**

The `length_of_section` specifies a measure of the longitudinal extent of the Section.

NOTE If more than one `Single_value` (see 4.2.316) object is assigned different aspects of the section's length are specified.

EXAMPLE To define a particular cabletray the typical, maximum, and minimum value of the `length_of_section` are specified.

See 4.3.2084 for the application assertion.

**4.2.297.8 space\_factor**

The `space_factor` specifies the ratio between the usable cross section to the geometrical cross section of the Section.

## ISO 10303-212:2001(E)

NOTE If more than one Single\_value (see 4.2.316) object is assigned different aspects of the section's space factor are specified.

EXAMPLE To define a particular cabletray the typical and minimum value of the space\_factor are specified.

See 4.3.2085 for the application assertion.

### 4.2.297.9 version\_id

The version\_id specifies versioning information for the Section.

The version\_id need not be specified for a particular Section.

### 4.2.298 Section\_end

The Section\_end specifies one extremity of the associated Section (see 4.2.296).

The data associated with a Section\_end are the following:

- id;
- kind;
- located\_at;
- of\_section.

#### 4.2.298.1 id

The id specifies the identifier of the Section\_end.

#### 4.2.298.2 kind

The kind specifies the type of the Section\_end. The value is either user defined or predefined.

The predefined value of kind is one of the following:

- flat oval;
- round;
- u shape.

NOTE See 4.2.298.2.1 - 4.2.298.2.3 for the definition of each predefined value for kind.

##### 4.2.298.2.1 flat oval

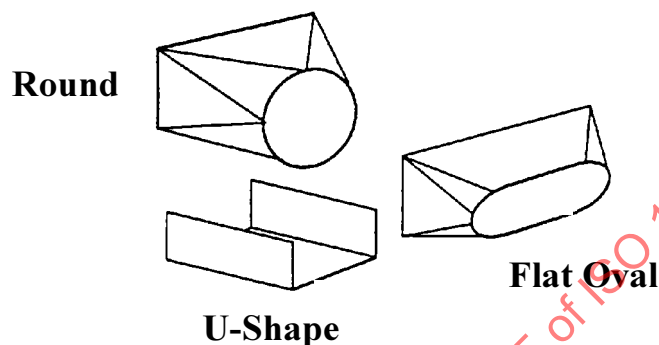
flat oval: The cross section of the Section\_end is as outlined in Figure 17.

**4.2.298.2.2 round**

round: The cross section of the Section\_end is as outlined in Figure 17.

**4.2.298.2.3 u shape**

u shape: The cross section of the Section\_end is as outlined in Figure 17.



**Figure 17 - Predefined section end shapes**

The kind need not be specified for a particular Section\_end.

**4.2.298.3 located\_at**

The located\_at specifies the position of the Section\_end.

The located\_at need not be specified for a particular Section\_end.

See 4.3.2086 for the application assertion.

**4.2.298.4 of\_section**

The of\_section specifies the Section (see 4.2.296) to which the Section\_end belongs.

See 4.3.2087 for the application assertion.

**4.2.299 Section\_interface**

A Section\_interface is a means to join Section (see 4.2.296) objects to each other.

## ISO 10303-212:2001(E)

The data associated with a Section\_interface are the following:

- cross\_sectional\_area;
- description;
- id;
- implemented\_by;
- joins;
- located\_at;
- space\_factor;
- version\_id.

### 4.2.299.1 cross\_sectional\_area

The cross\_sectional\_area specifies the transverse section of a Section\_interface in which the cabling can be placed.

See 4.3.2088 for the application assertion.

### 4.2.299.2 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Section\_interface.

The description need not be specified for a particular Section\_interface.

### 4.2.299.3 id

The id specifies the identifier of the Section\_interface.

### 4.2.299.4 implemented\_by

The implemented\_by specifies the equipment that realizes the Section\_interface.

See 4.3.2089 for the application assertion.

### 4.2.299.5 joins

The joins specifies the Section\_end (see 4.2.297) objects that are linked through the Section\_interface.

See 4.3.2091 for the application assertion.

### 4.2.299.6 located\_at

The located\_at specifies the position of the Section\_interface.



The located\_at need not be specified for a particular Section\_interface.

See 4.3.2090 for the application assertion.

#### **4.2.299.7 space\_factor**

The space\_factor specifies the ratio between the actually useable cross section to the geometrical cross section.

See 4.3.2092 for the application assertion.

#### **4.2.299.8 version\_id**

The version\_id specifies versioning information for the Section\_interface.

The version\_id need not be specified for a particular Section\_interface.

### **4.2.300 Section\_interface\_relationship**

A Section\_interface\_relationship is the relation between two Section\_interface (see 4.2.298) objects.

The data associated with a Section\_interface\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### **4.2.300.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Section\_interface\_relationship.

The description need not be specified for a particular Section\_interface\_relationship.

#### **4.2.300.2 related**

The related specifies the second of the two Section\_interface (see 4.2.298) objects related by the Section\_interface\_relationship.

See 4.3.2093 for the application assertion.

#### **4.2.300.3 relating**

The relating specifies the first of the two Section\_interface (see 4.2.298) objects related by the Section\_interface\_relationship.

See 4.3.2094 for the application assertion.

#### 4.2.300.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution;
- version hierarchy;
- version sequence.

NOTE 1 See 4.2.300.4.1 - 4.2.300.4.6 for the definition of each predefined value for relation\_type.

##### 4.2.300.4.1 alternate

alternate: The Section\_interface\_relationship defines a relationship where the related Section\_interface (see 4.2.298) is a possible substitute to the relating Section\_interface (see 4.2.298).

NOTE 2 This concept refers to the possibility to replace the Section\_interface (see 4.2.298). The actual replacement is addressed by 'substitution'.

##### 4.2.300.4.2 decomposition

decomposition: The Section\_interface\_relationship defines a relationship where the related Section\_interface (see 4.2.298) is one of the components into which the relating Section\_interface (see 4.2.298) is divided.

##### 4.2.300.4.3 derivation

derivation: The Section\_interface\_relationship defines a deriving relationship where the related Section\_interface (see 4.2.298) is based on the relating Section\_interface (see 4.2.298).

##### 4.2.300.4.4 substitution

substitution: The Section\_interface\_relationship defines a relationship where the related Section\_interface (see 4.2.298) replaces the relating Section\_interface (see 4.2.298).

##### 4.2.300.4.5 version hierarchy

version hierarchy: The Section\_interface\_relationship defines a hierarchical relationship where the related Section\_interface (see 4.2.298) is a subversion of the relating Section\_interface (see 4.2.298).

EXAMPLE 1 Revisions 1.1 and 1.2 of a Section\_interface (see 4.2.298).

**4.2.300.4.6 version sequence**

version sequence: The `Section_interface_relationship` defines a relationship where the relating `Section_interface` (see 4.2.298) is the preceding version and the related `Section_interface` (see 4.2.298) is the following version.

NOTE 3 The relationship does not imply inheritance of any kind between the application objects that are related.

EXAMPLE 2 'version sequence' is used, whenever a new version is prepared, e.g., 'version 1.0' is the preceding version for the following 'version 2.0'.

**4.2.301 Section\_relationship**

A `Section_relationship` is a relation between two `Section` (see 4.2.296) objects.

The data associated with an `Section_relationship` are the following:

- description;
- related;
- relating;
- relation\_type.

**4.2.301.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the `Section_relationship`.

The description need not be specified for a particular `Section_relationship`.

**4.2.301.2 related**

The related specifies the second of the two `Section` (see 4.2.296) objects related by the `Section_relationship`.

See 4.3.2095 for the application assertion.

**4.2.301.3 relating**

The relating specifies the first of the two `Section` (see 4.2.296) objects related by the `Section_relationship`.

See 4.3.2096 for the application assertion.

**4.2.301.4 relation\_type**

The `relation_type` specifies the meaning of the relationship. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of `relation_type` is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution;
- version hierarchy;
- version sequence.

NOTE 1 See 4.2.301.4.1 - 4.2.301.4.6 for the definition of each predefined value for `relation_type`.

### 4.2.301.4.1 alternate

alternate: The `Section_relationship` defines a relationship where the related Section (see 4.2.296) is a possible substitute to the relating Section (see 4.2.296).

NOTE 2 This concept refers to the possibility to replace the Section (see 4.2.296). The actual replacement is addressed by 'substitution'.

### 4.2.301.4.2 decomposition

decomposition: The `Section_relationship` defines a relationship where the related Section (see 4.2.296) is one of the components into which the relating Section (see 4.2.296) is divided.

### 4.2.301.4.3 derivation

derivation: The `Section_relationship` defines a deriving relationship where the related Section (see 4.2.296) is based on the relating Section (see 4.2.296).

### 4.2.301.4.4 substitution

substitution: The `Section_relationship` defines a relationship where the related Section (see 4.2.296) replaces the relating Section (see 4.2.296).

### 4.2.301.4.5 version hierarchy

version hierarchy: The `Section_relationship` defines a hierarchical relationship where the related Section (see 4.2.296) is a subversion of the relating Section (see 4.2.296).

EXAMPLE Revisions 1.1 and 1.2 of a Section (see 4.2.296).

### 4.2.301.4.6 version sequence

version sequence: The `Section_relationship` defines a succession of versions where the relating Section (see 4.2.296) is the preceding version, and the related Section (see 4.2.296) is the following version.

## 4.2.302 Security\_classification

A Security\_classification is a means to specify a security level for an physical or abstract item.

The data associated with a Security\_classification are the following:

- is\_applied\_to;
- name;
- purpose;
- security\_classification\_level.

### 4.2.302.1 is\_applied\_to

The is\_applied\_to specifies the item to which the Security\_classification is assigned.

Each is\_applied\_to may be one of the following: Activity (see 4.2.1), Activity\_element (see 4.2.2), Activity\_method\_assignment (see 4.2.4), Activity\_relationship (see 4.2.5), Alternate\_item\_relationship (see 4.2.11), Assembly\_component\_relationship (see 4.2.26), Assembly\_substitute\_relationship (see 4.2.28), Cable\_pull\_information (see 4.2.33), Class\_structure\_relationship (see 4.2.45), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_allocation (see 4.2.60), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Contract (see 4.2.63), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see 4.2.124), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Functional\_unit\_allocation (see 4.2.154), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Manufacturing\_configuration (see 4.2.198), Marking (see 4.2.199), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Offered\_function\_allocation (see 4.2.220), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Port\_allocation (see 4.2.248), Preferred\_item\_allocation (see 4.2.258), Preferred\_item\_terminal\_allocation (see 4.2.259), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Project (see 4.2.271), Requirement (see 4.2.285), Route (see 4.2.290), Route\_relationship (see 4.2.291), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Technical\_system (see 4.2.336), Technical\_system\_

## ISO 10303-212:2001(E)

relationship (see 4.2.337), Terminal (see 4.2.338), Work\_order (see 4.2.364), or Work\_request (see 4.2.365).

See 4.3.2097, 4.3.2098, 4.3.2099, 4.3.2100, 4.3.2101, 4.3.2102, 4.3.2103, 4.3.2104, 4.3.2105, 4.3.2106, 4.3.2107, 4.3.2108, 4.3.2109, 4.3.2110, 4.3.2111, 4.3.2112, 4.3.2113, 4.3.2114, 4.3.2115, 4.3.2116, 4.3.2117, 4.3.2118, 4.3.2119, 4.3.2120, 4.3.2121, 4.3.2122, 4.3.2123, 4.3.2124, 4.3.2125, 4.3.2126, 4.3.2127, 4.3.2128, 4.3.2129, 4.3.2130, 4.3.2131, 4.3.2132, 4.3.2133, 4.3.2134, 4.3.2135, 4.3.2136, 4.3.2137, 4.3.2138, 4.3.2139, 4.3.2140, 4.3.2141, 4.3.2142, 4.3.2143, 4.3.2144, 4.3.2145, 4.3.2146, 4.3.2147, 4.3.2148, 4.3.2149, 4.3.2150, 4.3.2151, 4.3.2152, 4.3.2153, 4.3.2154, 4.3.2155, 4.3.2156, 4.3.2157, 4.3.2158, 4.3.2159, 4.3.2160, 4.3.2161, 4.3.2162, 4.3.2163, 4.3.2164, 4.3.2165, 4.3.2166, 4.3.2167, 4.3.2168, 4.3.2169, 4.3.2170, 4.3.2171, 4.3.2172, 4.3.2173, 4.3.2174, 4.3.2175, 4.3.2176, 4.3.2177, 4.3.2178, 4.3.2179, 4.3.2180, 4.3.2181, 4.3.2182, 4.3.2184, 4.3.2185, 4.3.2186, 4.3.2187, 4.3.2188, 4.3.2189, 4.3.2190, 4.3.2191, 4.3.2192, and 4.3.2193 for the application assertions.

### 4.2.302.2 name

The name specifies the identifier of the Security\_classification.

### 4.2.302.3 purpose

The purpose specifies the rationale behind the Security\_classification.

### 4.2.302.4 security\_classification\_level

The security\_classification\_level specifies the Security\_level (see 4.2.302) associated with the item.

See 4.3.2183 for the application assertion.

## 4.2.303 Security\_level

A Security\_level is the specification of a level of security within some Security\_classification (see 4.2.301).

NOTE The values of Security\_level are company specific.

The data associated with a Security\_level are the following:

- level\_name;
- used\_classification\_system.

### 4.2.303.1 level\_name

The level\_name specifies the word or abbreviation used to refer to the Security\_level.

### 4.2.303.2 used\_classification\_system

The used\_classification\_system specifies the Classification\_system (see 4.2.48) that contains the information about how to interpret the name of the Security\_level.

The used\_classification\_system need not be specified for a particular Security\_level.

See 4.3.2194 for the application assertion.

#### 4.2.304 Selected\_device

A Selected\_device is a type of Device (see 4.2.88) that is the occurrence of a Design\_discipline\_item\_definition (see 4.2.86) which depends on certain constraints.

EXAMPLE To minimize the concentricity error of a wheel, weights are attached to the rim; this set of weights is a Selected\_device. The quantity, as well as the position of these weights, depends on the concentricity behaviour of each particular wheel as manufactured and tested.

The data associated with a Selected\_device are the following:

- selected\_quantity;
- selection\_control.

##### 4.2.304.1 selected\_quantity

The selected\_quantity specifies the number of occurrences foreseen as Selected\_device.

NOTE If the quantity is to be specified as a minimum or a maximum, then the Value\_limit (see 4.2.358) object may be used.

See 4.3.2195 for the application assertion.

##### 4.2.304.2 selection\_control

The selection\_control specifies the constraint that has to be evaluated for the Selected\_device.

EXAMPLE The information 'balancing the wheel' is an example for a selection\_control.

#### 4.2.305 Serial\_configuration

A Serial\_configuration is a type of Manufacturing\_configuration (see 4.2.198) that is planned to apply from a given serial number of the product for whose characteristic the Serial\_configuration identifies a solution.

The data associated with a Serial\_configuration are the following:

- serial\_end\_number;
- serial\_start\_number.

##### 4.2.305.1 serial\_end\_number

The serial\_end\_number specifies the serial number of that instance of the product that is the last instance for which the Serial\_configuration applies.

The serial\_end\_number need not be specified for a particular Serial\_configuration.

#### **4.2.305.2 serial\_start\_number**

The serial\_start\_number specifies the serial number of that instance of the product that is the first instance for which the Serial\_configuration applies.

#### **4.2.306 Set\_of\_notes**

A Set\_of\_notes is a type of Generic\_note (see 4.2.159) that is a collection of Note (see 4.2.210) objects.

The data associated with a Set\_of\_notes are the following:

— grouped\_notes.

##### **4.2.306.1 grouped\_notes**

The grouped\_notes specifies the Generic\_note (see 4.2.159) objects that are collected by a Set\_of\_notes.

See 4.3.2196 for the application assertion.

#### **4.2.307 Shape**

A Shape is the reference to the specification of the external form of a piece of equipment.

The data associated with a Shape are the following:

— id.

##### **4.2.307.1 id**

The id specifies the identifier of the Shape.

#### **4.2.308 Shape\_assignment**

A Shape\_assignment is a relation that associates a Shape (see 4.2.306) with an item.

NOTE If a shape is assigned to a functional item, the shape specifies the volume of space required to provide this service.

The data associated with a Shape\_assignment are the following:

— assigned\_shape;

— associated\_item;

— description;

— role.



### 4.2.308.1 assigned\_shape

The assigned\_shape specifies the Shape (see 4.2.306) object assigned to the item.

See 4.3.2211 for the application assertion.

### 4.2.308.2 associated\_item

The associated\_item specifies the item with which the Shape (see 4.2.306) is associated.

Each associated\_item may be one of the following: Connectivity\_definition (see 4.2.61), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Function\_definition (see 4.2.145), Function\_unit (see 4.2.148), Functional\_connectivity\_definition (see 4.2.152), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Location (see 4.2.192), Physical\_instance (see 4.2.243), Port (see 4.2.247), Product\_component (see 4.2.265), Section (see 4.2.296), Section\_interface (see 4.2.298), Technical\_system (see 4.2.336), or Terminal (see 4.2.338).

See 4.3.2197, 4.3.2198, 4.3.2199, 4.3.2200, 4.3.2201, 4.3.2202, 4.3.2203, 4.3.2204, 4.3.2205, 4.3.2206, 4.3.2207, 4.3.2208, 4.3.2209, 4.3.2210, 4.3.2212, and 4.3.2213 for the application assertions.

### 4.2.308.3 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Shape\_assignment.

The description need not be specified for a particular Shape\_assignment.

### 4.2.308.4 role

The role specifies the relationship between the item and the assigned shape. The value is either user defined or predefined.

The predefined value of role is one of the following:

- body shape;
- mounting space;
- safety space.

NOTE See 4.2.308.4.1 - 4.2.308.4.3 for the definition of each predefined value for relation\_type.

#### 4.2.308.4.1 body shape

body shape: The Shape (see 4.2.306) specifies the form of the associated item.

#### 4.2.308.4.2 mounting space

mounting space: The Shape (see 4.2.306) specifies the space required to put the item into position.

#### 4.2.308.4.3 safety space

safety space: The Shape (see 4.2.306) specifies the form of the safety zone required for the associated item.

#### 4.2.309 Sheet\_placed\_annotation

A Sheet\_placed\_annotation is a type of Draughting\_annotation (see 4.2.116) that is located in the coordinate system of the drawing sheet.

The data associated with a Sheet\_placed\_annotation are the following:

- annotation\_layers;
- annotation\_visibility.

##### 4.2.309.1 annotation\_layers

The annotation\_layers specifies the layers that contain the annotation.

See 4.3.2214 for the application assertion.

##### 4.2.309.2 annotation\_visibility

The annotation\_visibility specifies whether or not each piece of annotation placed within the drawing sheet is visible.

See 4.3.2215 for the application assertion.

#### 4.2.310 Signal

A Signal is the physical representation of a message or an information flow being generated, processed, or conveyed within an electrotechnical system.

NOTE 1 To be processed by an electrotechnical system a Process\_variable (see 4.2.260) needs to be converted into a Signal.

EXAMPLE 1 A temperature sensor converts the temperature of a fluid into an electrical current that is represented by a Signal. The Signal thus informs about the height of the temperature.

NOTE 2 By associating Signal\_value (see 4.2.313) objects to the Signal specific values of a Process\_variable (see 4.2.260) can be assigned to characteristic values of the Signal.

EXAMPLE 2 A temperature of 100 °C is represented by a Signal\_value (see 4.2.313) of 5 V.

NOTE 3 By associating General\_classification (see 4.2.156) objects to Signal the type of the Signal can be specified.

EXAMPLE 3 A Signal is categorized to be a 32 bit wide and to be of type 'input'.

The data associated with a Signal are the following:

- associated\_parameter;
- description;
- extended\_designation;
- id;
- signal\_level\_indicator;
- version\_id.

#### **4.2.310.1 associated\_parameter**

The associated\_parameter specifies the process variable about which the Signal informs.

See 4.3.2216 for the application assertion.

#### **4.2.310.2 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Signal.

The description need not be specified for a particular Signal.

#### **4.2.310.3 extended\_designation**

The extended\_designation specifies a structured label for the Signal.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

EXAMPLE IEC 61175 specifies designations for signals and connections.

## ISO 10303-212:2001(E)

The extended\_designation need not be specified for a particular Signal.

See 4.3.2217 for the application assertion.

### 4.2.310.4 id

The id specifies the identifier of the Signal.

### 4.2.310.5 signal\_level\_indicator

The signal\_level\_indicator specifies the logical state of the signal that carries the information.

The signal\_level\_indicator need not be specified for a particular Signal.

### 4.2.310.6 version\_id

The version\_id specifies versioning information for the Signal.

The version\_id need not be specified for a particular Signal.

## 4.2.311 Signal\_designation

A Signal\_designation is a type of Object\_designation (see 4.2.217) that is a reference designation which uniquely identifies a signal within its scope.

## 4.2.312 Signal\_relationship

A Signal\_relationship is the relation between two Signal (see 4.2.309) objects.

The data associated with a Signal\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.312.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Signal\_relationship.

The description need not be specified for a particular Signal\_relationship.

### 4.2.312.2 related

The related specifies the second of the two Signal (see 4.2.309) objects related by the Signal\_relationship.

See 4.3.2218 for the application assertion.

### 4.2.312.3 relating

The relating specifies the first of the two Signal (see 4.2.309) objects related by the Signal\_ - relationship.

See 4.3.2219 for the application assertion.

### 4.2.312.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution;
- redundancy;
- version hierarchy.

NOTE 1 See 4.2.312.4.1 - 4.2.312.4.6 for the definition of each predefined value for relation\_ - type.

#### 4.2.312.4.1 alternate

alternate: The Signal\_relationship defines a relationship where the related Signal (see 4.2.309) is a possible substitute to the relating Signal (see 4.2.309).

NOTE 2 This concept refers to the possibility to replace the Signal (see 4.2.309). The actual replacement is addressed by 'substitution'.

#### 4.2.312.4.2 decomposition

decomposition: The Signal\_relationship defines a relationship where the related Signal (see 4.2.309) is one of the components into which the relating Signal (see 4.2.309) is divided.

NOTE In the case in which a signal acts as a carrier for other signals, it can be decomposed into these signals.

#### 4.2.312.4.3 derivation

derivation: The Signal\_relationship defines a deriving relationship where the related Signal (see 4.2.309) is based on the relating Signal (see 4.2.309).

## ISO 10303-212:2001(E)

NOTE 1 A `Signal_relationship` with `relation_type` 'derivation' can be used to indicate the transfer function of a module. In such a case the input and output signal are associated through an instance of `Signal_relationship`. The 'description' attribute of that instance may contain the specification of the transfer function as a mathematical function.

NOTE 2 To avoid a shower of signals a summary signal can be derived from the cascade of signals that occur when a piece of equipment fails.

EXAMPLE A system used to keep a drive cool consists of a water pump, a radiator, and several sensors to monitor the system. An event that causes all of the water to leak out of the cooling system causes a number of signals. The temperature of the engine raises, the water pressure drops, and the emergency shutdown of the drive takes place. These three individual `Signal` (see 4.2.309) objects may be combined into a single `Signal` (see 4.2.309) object, which simply states that the drive has stopped, by relating the three `Signal` (see 4.2.309) objects to the summary `Signal` (see 4.2.309) through three `Signal_relationship` objects with `relation_type` 'derivation'.

### 4.2.312.4.4 substitution

substitution: The `Signal_relationship` defines a relationship where the related `Signal` (see 4.2.309) replaces the relating `Signal` (see 4.2.309).

### 4.2.312.4.5 redundancy

redundancy: The `Signal_relationship` defines a relationship where the related `Signal` (see 4.2.309) is replicated by the relating `Signal` (see 4.2.309).

EXAMPLE To provide for a fail-safe service a `Signal` (see 4.2.309) is replicated. If one `Signal` (see 4.2.309) is disturbed, the other is still readable.

### 4.2.312.4.6 version hierarchy

version hierarchy: The `Signal_relationship` defines a hierarchical relationship where the related `Signal` (see 4.2.309) is a subversion of the relating `Signal` (see 4.2.309).

EXAMPLE Revision 1.1 and 1.2 of a signal.

## 4.2.313 `Signal_system_assignment`

A `Signal_system_assignment` is a relation that associates a `Signal` (see 4.2.309) with a physical or abstract item that processes or transmits the `Signal` (see 4.2.309).

The data associated with a `Signal_system_assignment` are the following:

- `associated_signal`;
- `associated_system`;
- `description`;
- `role`.

#### **4.2.313.1 associated\_signal**

The `associated_signal` specifies the `Signal` (see 4.2.309).

See 4.3.2228 for the application assertion.

#### **4.2.313.2 associated\_system**

The `associated_system` specifies the item that processes or transmits the `Signal` (see 4.2.309).

Each `associated_system` may be one of the following: `Connectivity_definition` (see 4.2.61), `Design_discipline_item_definition` (see 4.2.86), `Device` (see 4.2.88), `Function_definition` (see 4.2.145), `Function_unit` (see 4.2.148), `Functional_connectivity_definition` (see 4.2.152), `Physical_instance` (see 4.2.243), `Port` (see 4.2.247), `Technical_system` (see 4.2.336), or `Terminal` (see 4.2.338).

See 4.3.2220, 4.3.2221, 4.3.2222, 4.3.2223, 4.3.2224, 4.3.2225, 4.3.2226, 4.3.2227, 4.3.2229, and 4.3.2230 for the application assertions.

#### **4.2.313.3 description**

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Signal_system_assignment`.

The `description` need not be specified for a particular `Signal_system_assignment`.

#### **4.2.313.4 role**

The `role` specifies the relationship between the signal and the associated system. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of role is one of the following:

- amplifier;
- monitor;
- source;
- target;
- transmitter.

NOTE See 4.2.313.4.1 - 4.2.313.4.5 for the definition of each predefined value for role.

### 4.2.313.4.1 amplifier

amplifier: The signal is boosted within the assigned system.

### 4.2.313.4.2 monitor

monitor: The signal is observed within the assigned system.

### 4.2.313.4.3 source

source: The assigned system acts as a source for the signal.

EXAMPLE A sensor can be a source for a signal.

### 4.2.313.4.4 target

target: The assigned system acts as a target for the signal.

EXAMPLE A screen that visualizes a message can be a target for a signal.

### 4.2.313.4.5 transmitter

transmitter: The signal is transferred within the assigned system.

## 4.2.314 Signal\_value

A Signal\_value is the measure of a signal.

NOTE 1 By association Signal\_value objects the characteristics of a Signal (see 4.2.309) can be laid down.

EXAMPLE 1 A Signal (see 4.2.309) can go from 0 to 10 mA with a linear characteristic.

NOTE 2 By associating Signal\_value objects to the Signal (see 4.2.309) specific values of a Process\_variable (see 4.2.260) can be assigned to characteristic values of the Signal (see 4.2.309).

EXAMPLE 2 A temperature of 100 °C is represented by a Signal\_value of 5 V.



The data associated with a `Signal_value` are the following:

- `associated_signal`;
- `characteristic`;
- `description`;
- `id`;
- `value_of_signal`;
- `valued_parameter`.

#### **4.2.314.1 associated\_signal**

The `associated_signal` specifies the `Signal` (see 4.2.309) to which the `Signal_value` applies.

See 4.3.2233 for the application assertion.

#### **4.2.314.2 characteristic**

The `characteristic` specifies how the `Signal_value` depends on the `Process_variable` (see 4.2.260) that is assigned through the `associated_parameter`. The value is either user defined or predefined.

The predefined value of `characteristic` is one of the following:

- `linear`.

NOTE 3 See 4.2.314.2.1 for the definition of each predefined value for `characteristic`.

##### **4.2.314.2.1 linear**

`linear`: The equation between the value of the `Process_variable` (see 4.2.260) and the `Signal_value` gives a straight line when plotted on a graph.

EXAMPLE 3 A temperature sensor with a linear transfer function will cause a linear dependency between the value of the associated `Process_variable` (see 4.2.260) characterizing the temperature and the associated `Signal_value`.

The `characteristic` need not be specified for a particular `Signal_value`.

#### **4.2.314.3 description**

The `description` specifies an alphanumerical string containing human-interpretable text that gives further details about the `Signal_value`.

The `description` need not be specified for a particular `Signal_value`.

#### **4.2.314.4 id**

The `id` specifies the identifier of the `Signal_value`.

#### **4.2.314.5 value\_of\_signal**

The `value_of_signal` specifies the numerical measure of the associated `Signal` (see 4.2.309).

See 4.3.2231 for the application assertion.

#### **4.2.314.6 valued\_parameter**

The `valued_parameter` specifies a `Process_variable` (see 4.2.260) object that corresponds to the `Signal_value`.

The `valued_parameter` need not be specified for a particular `Signal_value`.

See 4.3.2232 for the application assertion.

#### **4.2.315 Single\_device**

A `Single_device` is a type of `Device` (see 4.2.88) that is an individual piece of equipment.

#### **4.2.316 Single\_function\_unit**

A `Single_function_unit` is a type of `Function_unit` (see 4.2.148) that is an individual functional module.

#### **4.2.317 Single\_value**

A `Single_value` is a type of `Data_element_value` (see 4.2.76) that specifies a distinct value.

The data associated with a `Single_value` are the following:

— `value_of_single_value`.

##### **4.2.317.1 value\_of\_single\_value**

The `value_of_single_value` specifies the actual measure of the `Single_value`.

Each `value_of_single_value` may be one of the following: `Binary_value` (see 4.2.29), `Logical_value` (see 4.2.195), `String_value` (see 4.2.331), or `Value_with_unit` (see 4.2.360).

See 4.3.2234, 4.3.2235, 4.3.2236, and 4.3.2237 for the application assertions.

#### **4.2.318 Solid\_fill\_area**

A `Solid_fill_area` is a type of `Fill_area_appearance` (see 4.2.140) that uniformly fills the fill area to which the appearance is applied.

The data associated with a `Solid_fill_area` are the following:

— `fill_colour`.

##### **4.2.318.1 fill\_colour**

The fill\_colour specifies the colour definition that is uniformly applied to the fill area.

See 4.3.2238 for the application assertion.

### 4.2.319 Solution\_instance\_assignment

A Solution\_instance\_assignment is a relation that associates an Alternative\_solution (see 4.2.12) with the items that implement the solution.

The data associated with a Solution\_instance\_assignment are the following:

- instance;
- solution.

#### 4.2.319.1 instance

The instance specifies the item that implements the Alternative\_solution (see 4.2.12).

Each instance may be one of the following: Node (see 4.2.208), Notification (see 4.2.213), Path (see 4.2.232), Path\_node (see 4.2.233), Route (see 4.2.290), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Signal (see 4.2.309), or Technical\_system (see 4.2.336).

See 4.3.2240, 4.3.2241, 4.3.2242, 4.3.2243, 4.3.2244, 4.3.2245, 4.3.2246, 4.3.2247, 4.3.2248, and 4.3.2249 for the application assertions.

#### 4.2.319.2 solution

The solution specifies the Alternative\_solution (see 4.2.12) object.

See 4.3.2239 for the application assertion.

### 4.2.320 Specific\_document\_classification

A Specific\_document\_classification is a classification of a Document (see 4.2.101) with respect to specific criteria. The specific criteria are covered in the 'classification\_name' attribute.

The data associated with a Specific\_document\_classification are the following:

- associated\_document;
- classification\_name;
- description.

#### 4.2.320.1 associated\_document

The associated\_document the associated\_document specifies the Document (see 4.2.101) with which a particular Specific\_document\_classification is associated.

See 4.3.2250 for the application assertion.

## 4.2.320.2 classification\_name

The classification\_name provides classification information.

NOTE The overall classification information is obtained by traversing the hierarchical tree established by Specific\_item\_classification\_hierarchy (see 4.2.322).

The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- catalogue;
- manual;
- specification.

NOTE See 4.2.320.2.1 - 4.2.320.2.3 for the definition of each predefined value for relation\_type.

### 4.2.320.2.1 catalogue

catalogue: The assigned document is the catalogue in which the associated object is listed.

EXAMPLE The document can be the catalogue of the manufacturer.

### 4.2.320.2.2 manual

manual: The assigned document is the handbook that is supplied for the associated object.

### 4.2.320.2.3 specification

specification: The assigned document specifies the considerations that lead to the actual design of the associated object.

## 4.2.320.3 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Specific\_document\_classification.

The description need not be specified for a particular Specific\_document\_classification.

## 4.2.321 Specific\_document\_classification\_hierarchy

A Specific\_document\_classification\_hierarchy is used to build up hierarchical structures of Specific\_document\_classification\_hierarchy objects.

The data associated with a `Specific_document_classification` (see 4.2.319) are the following:

- `sub_classification`;
- `super_classification`.

#### **4.2.321.1 sub\_classification**

The `sub_classification` specifies the lower level of `Specific_document_classification` (see 4.2.319) in `Specific_document_classification_hierarchy` that is included in the super class.

See 4.3.2251 for the application assertion.

#### **4.2.321.2 super\_classification**

The `super_classification` specifies the higher level of `Specific_document_classification` (see 4.2.319) in `Specific_document_classification_hierarchy` that is included in the sub class.

See 4.3.2252 for the application assertion.

### **4.2.322 Specific\_item\_classification**

A `Specific_item_classification` is a classification of an `Item` (see 4.2.178) with respect to specific criteria. The specific criteria are covered in the '`classification_name`' attribute.

NOTE 1 If an `Item` (see 4.2.178) requires classification by more than one criterion, several `Specific_item_classification` objects are associated to the same `Item` (see 4.2.178).

NOTE 2 For the attribute '`classification_name`' a set of predefined values is specified hereafter. If values other than the proposed ones are used, they should be of general classifying nature. This kind of classification ought not to be used to classify names or identifiers of objects, e.g., in order to classify part families; for this purpose `General_classification` (see 4.2.156) ought to be used.

The data associated with a `Specific_item_classification` are the following:

- `associated_item`;
- `classification_name`;
- `description`.

#### **4.2.322.1 associated\_item**

The `associated_item` specifies the `Item` (see 4.2.178) with which a particular `Specific_item_classification` is associated.

See 4.3.2253 for the application assertion.

#### **4.2.322.2 classification\_name**

The `classification_name` provides classification information.

## ISO 10303-212:2001(E)

The following values shall be used:

- accessory;
- application control;
- assembly;
- collection;
- completely knocked down;
- detail;
- in process;
- part;
- prototype;
- raw material;
- regulated;
- safety;
- service;
- software;
- tool.

NOTE See 4.2.322.2.1 - 4.2.322.2.14 for the definition of each predefined value for classification\_name.

### 4.2.322.2.1 accessory

accessory: This type of classification is used to indicate that an Item (see 4.2.178) shall be considered to be a supplementary portion of an electrotechnical system with the obligation to help the system to perform its function.

### 4.2.322.2.2 application control

application control: This type of classification is used to indicate that an Item (see 4.2.178) shall be considered under certification aspects; these aspects may be specified further by the 'description' attribute.

EXAMPLE 1 Prior to the release of a new car to the market, both function and quality of certain parts have to be certified by an authority, e.g., the department of transportation. For such Item (see 4.2.178) objects, certification requirements have to be considered during the design phase.

### 4.2.322.2.3 assembly

assembly: This type of classification shall be used for any Item (see 4.2.178) that has an Assembly\_ - definition (see 4.2.27) provided for at least one of its versions, i.e., it is decomposed further.

#### **4.2.322.2.4 completely knocked down**

completely knocked down: This type of classification is used to indicate that an Item (see 4.2.178) is used in a production site that has assembling facilities only.

EXAMPLE 2 The 'completely knocked down' may indicate that the components are shipped to and assembled in a different country.

#### **4.2.322.2.5 detail**

detail: This type of classification shall be used for any Item (see 4.2.178) that has no Assembly\_ - definition (see 4.2.27) provided for any of its versions, i.e., it is not further decomposed.

#### **4.2.322.2.6 in process**

in process: This type of classification is used to indicate that the Item (see 4.2.178) identifies an intermediate object in a manufacturing process.

#### **4.2.322.2.7 part**

part: The Item (see 4.2.178) plays the role of a component of relevance within the system.

#### **4.2.322.2.8 prototype**

prototype: This type of classification is used to indicate that the Item (see 4.2.178) identifies a prototype and is not intended for serial production.

#### **4.2.322.2.9 raw material**

raw material: The Item (see 4.2.178) plays the role of raw material.

#### **4.2.322.2.10 regulated**

regulated: This type of classification is used to indicate that for an Item (see 4.2.178) certain regulations have to be considered.

#### **4.2.322.2.11 safety**

safety: This type of classification is used to indicate that an Item (see 4.2.178) is relevant for safety purposes.

#### **4.2.322.2.12 service**

service: This type of classification is used to indicate that an Item (see 4.2.178) is relevant for service purposes.

#### **4.2.322.2.13 software**

software: This type of classification is used to indicate that an Item (see 4.2.178) is a program that can be executed on a particular kind of computer.

#### 4.2.322.2.14 tool

tool: The Item (see 4.2.178) plays the role of a tool.

#### 4.2.322.3 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Specific\_item\_classification.

The description need not be specified for a particular Specific\_item\_classification.

#### 4.2.323 Specific\_item\_classification\_hierarchy

A Specific\_item\_classification\_hierarchy is used to build up hierarchical structures of Specific\_item\_classification (see 4.2.321). The top level in the hierarchy shall be a 'part' or 'software'.

The data associated with a Specific\_item\_classification\_hierarchy are the following:

- sub\_classification;
- super\_classification.

##### 4.2.323.1 sub\_classification

The sub\_classification specifies the lower level of Specific\_item\_classification (see 4.2.321) in a Specific\_item\_classification\_hierarchy that is included in the super class.

See 4.3.2254 for the application assertion.

##### 4.2.323.2 super\_classification

The super\_classification specifies the higher level of Specific\_item\_classification (see 4.2.321) in a Specific\_item\_classification\_hierarchy that includes the sub class.

See 4.3.2255 for the application assertion.

#### 4.2.324 Specification

A Specification is a characteristic of a product. It discriminates one product from other constituents of the same Product\_class (see 4.2.263). A Specification refers to a Specification\_category (see 4.2.324) that completes the semantic of the Specification.



NOTE 1 A Specification may be a characteristic of the members of more than one Product\_class (see 4.2.263) using Class\_specification\_association (see 4.2.44) objects.

NOTE 2 A Specification, in combination with a Configuration (see 4.2.56), can be used to define the conditions under which an Item (see 4.2.178) is used for a product of a Product\_class (see 4.2.263).

The data associated with a Specification are the following:

- category;
- description;
- id;
- name;
- package;
- version\_id.

#### **4.2.324.1 category**

The category specifies the Specification\_category (see 4.2.324) that completes the semantic of the Specification.

See 4.3.2256 for the application assertion.

#### **4.2.324.2 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Specification.

The description need not be specified for a particular Specification.

#### **4.2.324.3 id**

The id specifies the identifier of the Specification.

#### **4.2.324.4 name**

The name specifies a speaking designation of the Specification.

The name need not be specified for a particular Specification.

#### **4.2.324.5 package**

The package specifies whether or not this Specification represents a package of Specification objects. A package is a set of Specification objects that can be defined by the marketing department. A package combines those Specification objects that shall be offered to the market as a set. In the case where package is 'true', exactly one Specification\_inclusion (see 4.2.327) shall refer to this Specification as an if condition.

## ISO 10303-212:2001(E)

NOTE Usually the members of a package belong to distinct Specification\_category (see 4.2.324) objects.

### 4.2.324.6 version\_id

The version\_id specifies versioning information for the Specification.

The version\_id need not be specified for a particular Specification.

### 4.2.325 Specification\_category

A Specification\_category is the definition of a set of Specification (see 4.2.323) objects for the same purpose.

The data associated with a Specification\_category are the following:

- description;
- id;
- implicit\_exclusive\_condition.

#### 4.2.325.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Specification\_category.

#### 4.2.325.2 id

The id specifies the identifier of the Specification\_category.

#### 4.2.325.3 implicit\_exclusive\_condition

The implicit\_exclusive\_condition specifies whether or not the Specification (see 4.2.323) objects within the Specification\_category are mutually exclusive for the production of one particular product.

NOTE More complex conditions can be handled using Specification\_expression (see 4.2.326) objects.

### 4.2.326 Specification\_category\_hierarchy

A Specification\_category\_hierarchy is used to build up hierarchical structures of Specification\_category (see 4.2.324) objects.

The data associated with a Specification\_category\_hierarchy are the following:

- sub\_category;
- super\_category.

#### 4.2.326.1 sub\_category

The sub\_category is the lower level of Specification\_category (see 4.2.324) in Specification\_category\_hierarchy.

See 4.3.2257 for the application assertion.

#### **4.2.326.2 super\_category**

The super\_category is the higher level of Specification\_category (see 4.2.324) in Specification\_category\_hierarchy.

See 4.3.2258 for the application assertion.

### **4.2.327 Specification\_expression**

A Specification\_expression is a combination of Specification (see 4.2.323) objects formed by Boolean operations.

The data associated with a Specification\_expression are the following:

- description;
- id;
- operand;
- operation.

#### **4.2.327.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Specification\_expression.

The description need not be specified for a particular Specification\_expression.

#### **4.2.327.2 id**

The id specifies the identifier of the Specification\_expression.

The id need not be specified for a particular Specification\_expression.

#### **4.2.327.3 operand**

The operand specifies the operands of the Boolean operation that are either Specification (see 4.2.323) objects or other Specification\_expression objects.

Each operand may be one of the following: Specification (see 4.2.323) or Specification\_expression.

See 4.3.2259 and 4.3.2260 for the application assertions.

#### **4.2.327.4 operation**

The operation specifies the kind of Boolean operation.

## ISO 10303-212:2001(E)

The value of operation is one of the following:

- and;
- not;
- oneof;
- or.

NOTE See 4.2.327.4.1 - 4.2.327.4.4 for the definition of each permissible value for operation.

### 4.2.327.4.1 and

and: All of the identified Specification (see 4.2.323) objects shall be used.

### 4.2.327.4.2 not

not: The identified Specification (see 4.2.323) shall not be used.

### 4.2.327.4.3 oneof

oneof: Exactly one of the identified Specification (see 4.2.323) objects shall be used.

### 4.2.327.4.4 or

or: A subset or all of the identified Specification (see 4.2.323) objects shall be used.

## 4.2.328 Specification\_inclusion

A Specification\_inclusion is the representation of the statement that the application of a Specification (see 4.2.323) or a Specification\_expression (see 4.2.326) implies the inclusion of an additional Specification (see 4.2.323) or Specification\_expression (see 4.2.326).

NOTE The Specification\_inclusion is intended to complete the set of Specification (see 4.2.323) objects for a Product\_specification (see 4.2.269) in order to enable the manufacturing of the product on the basis of an initial set of Specification (see 4.2.323) objects defined, e.g., by a customer order.

The data associated with a Specification\_inclusion are the following:

- description;
- id;
- if\_condition;
- included\_specification.

### 4.2.328.1 description

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Specification\_inclusion.

The description need not be specified for a particular Specification\_inclusion.

#### 4.2.328.2 id

The id specifies the identifier of the Specification\_inclusion.

The id need not be specified for a particular Specification\_inclusion.

#### 4.2.328.3 if\_condition

The if\_condition specifies the Specification (see 4.2.323) or the Specification\_expression that serves as the condition for the inclusion.

Each if\_condition may be one of the following: Specification (see 4.2.323) or Specification\_expression (see 4.2.326).

See 4.3.2261 and 4.3.2263 for the application assertions.

#### 4.2.328.4 included\_specification

The included\_specification specifies the Specification (see 4.2.323) or the Specification\_expression (see 4.2.326) objects that are to be included.

NOTE In the case where the included\_specification is a Specification\_expression (see 4.2.326), i.e., an OR expression, several alternatives for Specification (see 4.2.323) may be included. In the case where more than one Specification (see 4.2.323) objects are to be included, a Specification\_expression (see 4.2.326) of type AND shall be used.

Each included\_specification may be one of the following: Specification (see 4.2.323) or Specification\_expression (see 4.2.326).

See 4.3.2262 and 4.3.2264 for the application assertions.

#### 4.2.329 Specified\_device

A Specified\_device is a type of Device (see 4.2.88) that is the mechanism to identify a certain Device (see 4.2.88) in a multi-level assembly structure that utilizes structural reuse of partial decompositions.

EXAMPLE A belt conveyor consists among others of the components front drive and rear drive. Both components contain a motor and some other components. In order to identify the motor of the front drive, the Specified\_device object references motor as related instance, front drive as upper\_usage, and belt conveyor as assembly\_context.

The object, which is referred by the 'definition' attribute of a Specified\_device and the object which is referenced through the 'related\_function\_unit' attribute as 'definition', shall be the same.

The data associated with a Specified\_device are the following:

## ISO 10303-212:2001(E)

- assembly\_context;
- related\_device;
- upper\_usage.

### 4.2.329.1 assembly\_context

The assembly\_context specifies an Assembly\_definition (see 4.2.27) object in which the instance is used that is identified by this mechanism.

The assembly\_context need not be specified for a particular Specified\_device.

See 4.3.2265 for the application assertion.

### 4.2.329.2 related\_device

The related\_device specifies the Device (see 4.2.88) that is to be identified. The related\_device shall not be of type Specified\_device.

See 4.3.2266 for the application assertion.

### 4.2.329.3 upper\_usage

The upper\_usage specifies the Device (see 4.2.88) in which the item referred by related\_instance occurs.

See 4.3.2267 for the application assertion.

### 4.2.330 Specified\_function\_unit

A Specified\_function\_unit is a type of Function\_unit (see 4.2.148) that is the mechanism to identify a certain Function\_unit (see 4.2.148) in a multi-level decomposition structure that utilizes structural reuse of partial decompositions.

**EXAMPLE** An audio-amplifier consists among others of the modules left-channel amplifier and right-channel amplifier. Both modules contain pre-amplifier and output amplifier. In order to identify the pre-amplifier of the left-channel amplifier, the Specified\_function\_unit object references pre-amplifier as related\_instance, left-channel amplifier as upper\_usage, and audio-amplifier as functional\_context.

The object, which is referred by the 'definition' attribute of a Specified\_function\_unit and the object which is referenced through the 'related\_function\_unit' attribute as 'definition', shall be the same.

The data associated with a Specified\_function\_unit are the following:

- functional\_context;
- related\_function\_unit;
- upper\_usage.

#### 4.2.330.1 functional\_context

The functional\_context specifies a Function\_unit (see 4.2.148) object in which the instance identified by this mechanism is used.

The functional\_context need not be specified for a particular Specified\_function\_unit.

See 4.3.2268 for the application assertion.

#### 4.2.330.2 related\_function\_unit

The related\_function\_unit specifies the Function\_unit (see 4.2.148) that is to be identified. The related\_function\_unit shall not be of type Specified\_function\_unit.

See 4.3.2269 for the application assertion.

#### 4.2.330.3 upper\_usage

The upper\_usage specifies the Function\_unit (see 4.2.148) in which the item referred by related\_function\_unit occurs.

See 4.3.2270 for the application assertion.

### 4.2.331 Storage\_temperature

A Storage\_temperature is the allowed ambient temperature of a component while the device is in storage.

NOTE Minimum and maximum of Storage\_temperature may be specified by assigning a Value\_range (see 4.2.359) object.

EXAMPLE The Storage\_temperature of a notebook computer ranges from -20°C to 60°C, whereas the Operating\_temperature (see 4.2.221) ranges from 5°C to 35°C.

The data associated with a Storage\_temperature are the following:

- temperature.

#### 4.2.331.1 temperature

The temperature specifies the value of the Storage\_temperature.

See 4.3.2271 for the application assertion.

### 4.2.332 String\_value

## ISO 10303-212:2001(E)

A `String_value` represents a sequence of one or more alphanumeric characters.

The data associated with a `String_value` are the following:

- `value_of_string_value`.

### 4.2.332.1 `value_of_string_value`

The `value_of_string_value` specifies the content of the `String_value`.

### 4.2.333 `Structured_dimension_callout`

A `Structured_dimension_callout` is a type of `Dimension_callout` (see 4.2.94) that is a callout wherein each component is identified as having the semantics of prefix information, suffix information, a dimension symbol, a dimension unit, a dimension value, or tolerance information.

The data associated with a `Structured_dimension_callout` are the following:

- `dimension_value`;
- `prefix_callout`;
- `suffix_callout`;
- `symbol`;
- `tolerance_value`;
- `unit_text`.

#### 4.2.333.1 `dimension_value`

The `dimension_value` specifies the text strings that present the actual value of the measurement.

See 4.3.2275 for the application assertion.

#### 4.2.333.2 `prefix_callout`

The `prefix_callout` specifies information used in interpreting the dimension or its applicability and is physically located before the dimension value as the dimension is read.

The `prefix_callout` need not be specified for a particular `Structured_dimension_callout`.

See 4.3.2273 for the application assertion.

#### 4.2.333.3 `suffix_callout`

The `suffix_callout` specifies information, physically located after the dimension value as the dimension is read, either used in interpreting the dimension or its applicability or used as additional information in conjunction with the dimension.

The `suffix_callout` need not be specified for a particular `Structured_dimension_callout`.



See 4.3.2274 for the application assertion.

#### **4.2.333.4 symbol**

The symbol specifies an Annotation\_symbol (see 4.2.20) that is used in conjunction with the dimension value to clarify the meaning of the measurement.

The symbol need not be specified for a particular Structured\_dimension\_callout.

See 4.3.2272 for the application assertion.

#### **4.2.333.5 tolerance\_value**

The tolerance\_value specifies the text strings that present the tolerance information for the measurement.

See 4.3.2276 for the application assertion.

#### **4.2.333.6 unit\_text**

The unit\_text specifies the text strings that present the unit of measurement.

See 4.3.2277 for the application assertion.

#### **4.2.334 Sub\_group**

A Sub\_group is a type of Group\_element (see 4.2.165) that is a collection of elements previously defined as a group.

The data associated with a Sub\_group are the following:

- basis\_group.

##### **4.2.334.1 basis\_group**

The basis\_group specifies the group that is used as a Sub\_group.

See 4.3.2278 for the application assertion.

#### **4.2.335 Supplier\_solution**

A Supplier\_solution is a type of Alternative\_solution (see 4.2.12) that is a solution where the supplier differs from other solutions for the same Product\_component (see 4.2.265).

The data associated with a Supplier\_solution are the following:

- probability\_rate;
- supplier.

##### **4.2.335.1 probability\_rate**

## ISO 10303-212:2001(E)

The `probability_rate` specifies the share that is assigned to a supplier in the context of the base element.

The `probability_rate` need not be specified for a particular `Supplier_solution`.

### 4.2.335.2 supplier

The supplier specifies the supplier for a `Supplier_solution`.

See 4.3.2279 for the application assertion.

### 4.2.336 Technical\_solution

A `Technical_solution` is a type of `Alternative_solution` (see 4.2.12) that is a solution where the functional requirements are fulfilled in a certain technical way.

The data associated with a `Technical_solution` are the following:

- description.

#### 4.2.336.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the `Technical_solution`.

### 4.2.337 Technical\_system

A `Technical_system` is the overall system that is to be planned, designed, or commissioned and that contains electrotechnical aspects. The `Technical_system` groups the physical and abstract items that, as a whole, make up the overall system.

EXAMPLE The power distribution system of a ship may be a `Technical_system`.

The data associated with a `Technical_system` are the following:

- contains;
- description;
- extended\_designation;
- id;
- version\_id.

#### 4.2.337.1 contains

The contains specifies the items that comprise the `Technical_system`.

Each contains may be one of the following: `Classification_system` (see 4.2.48), `Device` (see 4.2.88), `Document_representation` (see 4.2.110), `Function_unit` (see 4.2.148), `Notification` (see 4.2.213),

Physical\_instance (see 4.2.243), Process\_variable (see 4.2.260), Requirement (see 4.2.285), Route (see 4.2.290), or Signal (see 4.2.309).

See 4.3.2280, 4.3.2281, 4.3.2282, 4.3.2283, 4.3.2284, 4.3.2286, 4.3.2287, 4.3.2288, 4.3.2289, and 4.3.2290 for the application assertions.

#### **4.2.337.2 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Technical\_system.

The description need not be specified for a particular Technical\_system.

#### **4.2.337.3 extended\_designation**

The extended\_designation specifies a structured label for the Technical\_system.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

The extended\_designation need not be specified for a particular Technical\_system.

See 4.3.2285 for the application assertion.

#### **4.2.337.4 id**

The id specifies the identifier of the Technical\_system.

#### **4.2.337.5 version\_id**

The version\_id specifies versioning information for the Technical\_system.

The version\_id need not be specified for a particular Technical\_system.

### **4.2.338 Technical\_system\_relationship**

A Technical\_system\_relationship is the relation between two Technical\_system (see 4.2.336) objects.

The data associated with a Technical\_system\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

#### **4.2.338.1 description**

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Technical\_system\_relationship.

## ISO 10303-212:2001(E)

The description need not be specified for a particular Technical\_system\_relationship.

### 4.2.338.2 related

The related specifies the second of the two Technical\_system (see 4.2.336) objects related by the Technical\_system\_relationship.

See 4.3.2291 for the application assertion.

### 4.2.338.3 relating

The relating specifies the first of the two Technical\_system (see 4.2.336) objects related by the Technical\_system\_relationship.

See 4.3.2292 for the application assertion.

### 4.2.338.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

- alternate;
- decomposition;
- derivation;
- substitution;
- redundancy.

NOTE 1 See 4.2.338.4.1 - 4.2.338.4.5 for the definition of each predefined value for relation\_type.

#### 4.2.338.4.1 alternate

alternate: The Technical\_system\_relationship defines a relationship where the related Technical\_system (see 4.2.336) is a possible substitute to the relating Technical\_system (see 4.2.336).

NOTE 2 This concept refers to the possibility to replace the Technical\_system (see 4.2.336). The actual replacement is addressed by 'substitution'.

#### 4.2.338.4.2 decomposition

decomposition: The Technical\_system\_relationship defines a relationship where the related Technical\_system (see 4.2.336) is one of the components into which the relating Technical\_system (see 4.2.336) is divided.

**4.2.338.4.3 derivation**

derivation: The `Technical_system_relationship` defines a deriving relationship where the related `Technical_system` (see 4.2.336) is based on the relating `Technical_system` (see 4.2.336).

**4.2.338.4.4 substitution**

substitution: The `Technical_system_relationship` defines a relationship where the related `Technical_system` (see 4.2.336) replaces the relating `Technical_system` (see 4.2.336).

**4.2.338.4.5 redundancy**

redundancy: The `Technical_system_relationship` defines a relationship where the related `Technical_system` (see 4.2.336) is replicated by the relating `Technical_system` (see 4.2.336).

EXAMPLE To provide for a fail-safe service a `Technical_system` (see 4.2.336) is replicated. If one `Technical_system` (see 4.2.336) fails, the other is still in service.

**4.2.339 Terminal**

A Terminal is the occurrence of an `Interface_terminal` (see 4.2.174) used to access a piece of equipment.

The data associated with a Terminal are the following:

- `associated_interface_terminal`;
- `description`;
- `extended_designation`;
- `id`;
- `implemented_by`;
- `terminal_of`.

**4.2.339.1 associated\_interface\_terminal**

The `associated_interface_terminal` specifies the `Interface_terminal` (see 4.2.174) that defines the access to the piece of equipment.

See 4.3.2294 for the application assertion.

**4.2.339.2 description**

The `description` specifies an alphanumeric string containing human-interpretable text that gives further details about the Terminal.

The `description` need not be specified for a particular Terminal.

### 4.2.339.3 extended\_designation

The extended\_designation specifies a structured label for the Terminal.

NOTE The label assigned through extended\_designation shall be identical to the label assigned by the 'id' attribute.

The extended\_designation need not be specified for a particular Terminal.

See 4.3.2297 for the application assertion.

### 4.2.339.4 id

The id specifies the identifier of the Terminal.

### 4.2.339.5 implemented\_by

The implemented\_by specifies the Device (see 4.2.88) objects that are used to implement the Terminal.

Each implemented\_by may be one of the following: Device (see 4.2.88) or Physical\_instance (see 4.2.243).

See 4.3.2293 and 4.3.2295 for the application assertions.

### 4.2.339.6 terminal\_of

The terminal\_of specifies the Single\_device (see 4.2.314) to which the Terminal belongs.

See 4.3.2296 for the application assertion.

## 4.2.340 Terminal\_relationship

A Terminal\_relationship is the relation between two Terminal (see 4.2.338) objects.

The data associated with an Terminal\_relationship are the following:

- description;
- related;
- relating;
- relation\_type.

### 4.2.340.1 description

The description specifies an alphanumerical string containing human-interpretable text that gives further details about the Terminal\_relationship.

The description need not be specified for a particular Terminal\_relationship.

#### **4.2.340.2 related**

The related specifies the second of the two Terminal (see 4.2.338) objects related by the Terminal\_ - relationship.

See 4.3.2298 for the application assertion.

#### **4.2.340.3 relating**

The relating specifies the first of the two Terminal (see 4.2.338) objects related by the Terminal\_ - relationship.

See 4.3.2299 for the application assertion.

#### 4.2.340.4 relation\_type

The relation\_type specifies the meaning of the relationship. The value is either user defined or predefined.

The predefined value of relation\_type is one of the following:

— decomposition;

— redundancy.

NOTE See 4.2.340.4.1 and 4.2.340.4.2 for the definition of each predefined value for relation\_type.

##### 4.2.340.4.1 decomposition

decomposition: The Terminal\_relationship defines a relationship where the related Terminal (see 4.2.338) is one of the components into which the relating Terminal (see 4.2.338) is broken down.

##### 4.2.340.4.2 redundancy

redundancy: The Terminal\_relationship defines a relationship where the related Terminal (see 4.2.338)(see 4.2.156) is replicated by the relating Terminal (see 4.2.338).

EXAMPLE To provide for a fail-safe service a Terminal (see 4.2.338) is replicated. If one Terminal (see 4.2.338) fails, the other is still in service.

#### 4.2.341 Terminal\_designation

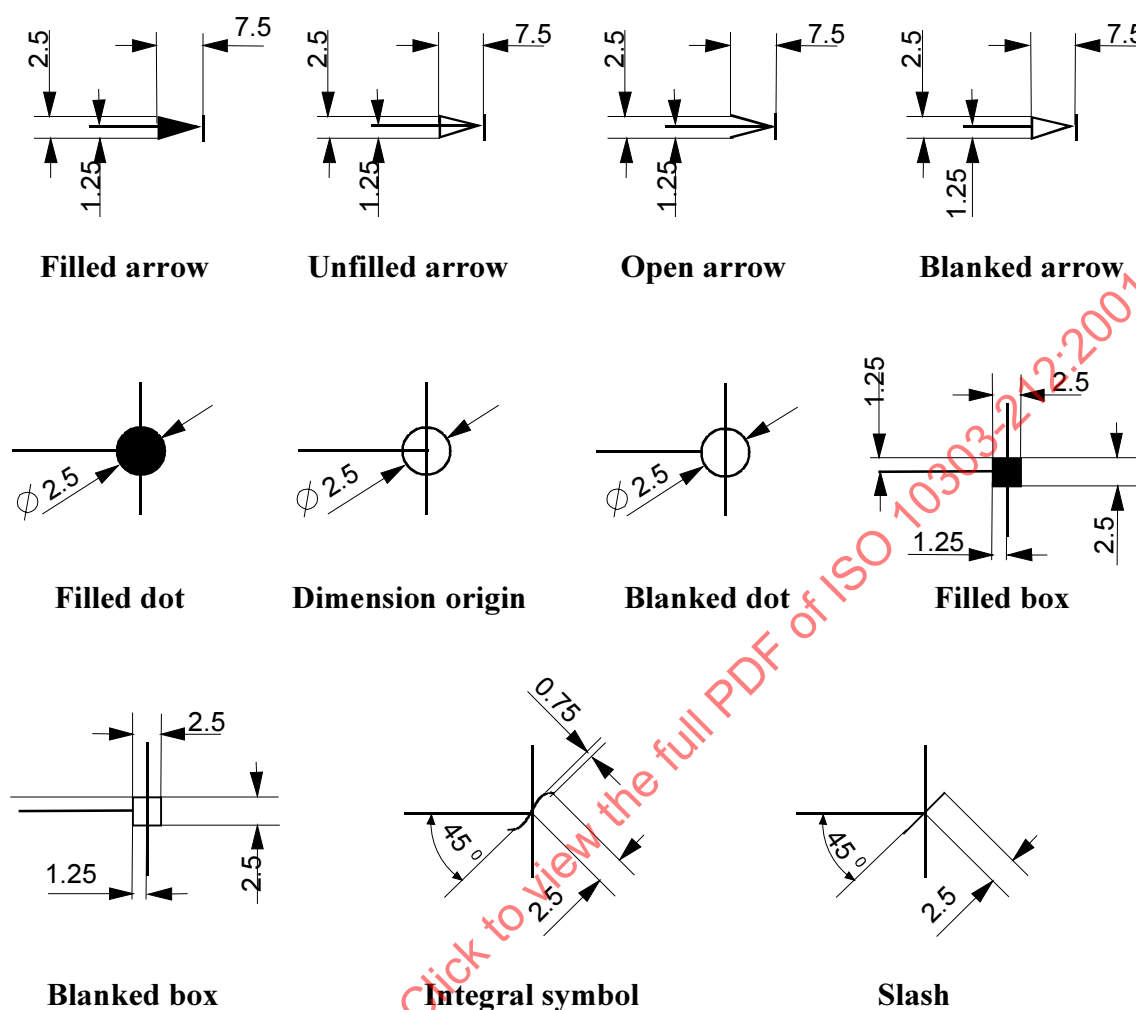
A Terminal\_designation is a type of Object\_designation (see 4.2.217) that is a reference designation to identify an access node with respect to the function or product to which it belongs.

#### 4.2.342 Terminator\_symbol

A Terminator\_symbol is a type of Predefined\_symbol (see 4.2.255) that is applied to an annotation curve and used to identify the endpoint or Point of application of any annotation directed by that curve.

The predefined terminator symbols that shall be supported by all implementations of this part of ISO 10303 are shown in Figure 18. The orientation of the individual symbol, as positioned on a horizontal dimension line, is shown in Figure 18.





**Figure 18 - Predefined terminator symbols**

The data associated with a Terminator\_symbol are the following:

— symbol\_type.

#### 4.2.342.1 symbol\_type

The symbol\_type specifies an alphanumerical string identifying the Terminator\_symbol in accordance with the definitions given below.

## ISO 10303-212:2001(E)

The value of symbol\_type is one of the following:

- blanked arrow;
- blanked box;
- blanked dot;
- dimension origin;
- filled arrow;
- filled box;
- filled dot;
- integral symbol;
- open arrow;
- slash;
- unfilled arrow;
- unfilled dot.

NOTE See 4.2.342.1.1 - 4.2.342.1.12 for the definition of each permissible value for symbol\_type.

### 4.2.342.1.1 blanked arrow

blanked arrow: The Terminator\_symbol is depicted as three line segments that form an isosceles triangle. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve to which the symbol is applied acts as a bisector to the angle created by the two equal sides. The area within the symbol is blanked. The size and graphical representation of the symbol are shown in Figure 18.

### 4.2.342.1.2 blanked box

blanked box: The Terminator\_symbol is depicted as four line segments that form a rectangle. The origin of the symbol is the geometric centre of the rectangle. The area within the symbol is blanked. The size and graphical representation of the symbol are shown in Figure 18.

### 4.2.342.1.3 blanked dot

blanked dot: The Terminator\_symbol is depicted as a circle. The origin of the symbol is the centre of the circle. The area within the symbol is blanked. The size and graphical representation of the symbol are shown in Figure 18.

### 4.2.342.1.4 dimension origin

dimension origin: The Terminator\_symbol is depicted as a circle. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.5 filled arrow**

filled arrow: The Terminator\_symbol is depicted as three line segments that form an isosceles triangle. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve to which the symbol is applied, acts as a bisector to the angle created by the two equal sides. The area within the symbol is shaded. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.6 filled box**

filled box: The Terminator\_symbol is depicted as four line segments that form a rectangle. The origin of the symbol is the geometric centre of the rectangle. The area within the symbol is shaded. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.7 filled dot**

filled dot: The Terminator\_symbol is depicted as a circle. The origin of the symbol is the centre of the circle. The area within the symbol is shaded. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.8 integral symbol**

integral symbol: The Terminator\_symbol is depicted as one line segment forming two adjacent arcs. The origin of the symbol is the midpoint between the two arcs. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.9 open arrow**

open arrow: The Terminator\_symbol is depicted as three line segments that form an isosceles triangle where the third side of the triangle is blanked. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve to which the symbol is applied, acts as a bisector to the angle created by the two equal sides. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.10 slash**

slash: The Terminator\_symbol is depicted as a line segment with the midpoint of the segment being the origin and lying on the annotation curve to which it is applied. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.11 unfilled arrow**

unfilled arrow: The Terminator\_symbol is depicted as three line segments that form an isosceles triangle. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve to which the symbol is applied, acts as a bisector to the angle created by the two equal sides. The size and graphical representation of the symbol are shown in Figure 18.

**4.2.342.1.12 unfilled dot**

unfilled dot: The Terminator\_symbol is depicted as a circle. The origin of the symbol is the centre of the circle. The size and graphical representation of the symbol are shown in Figure 18.

### 4.2.343 Text

A Text is a type of Annotation\_element (see 4.2.15) that is a collection of characters that convey some human-interpretable information.

The data associated with a Text are the following:

- alignment;
- blanking\_box;
- boundary\_of\_displayed\_box;
- default\_appearance;
- language\_code;
- mirror\_angle;
- surrounding\_box.

NOTE See Figure 19 for an illustration of the blanking box of text, the location and rotation of a text string, the alignment of text, and the surrounding box of text.

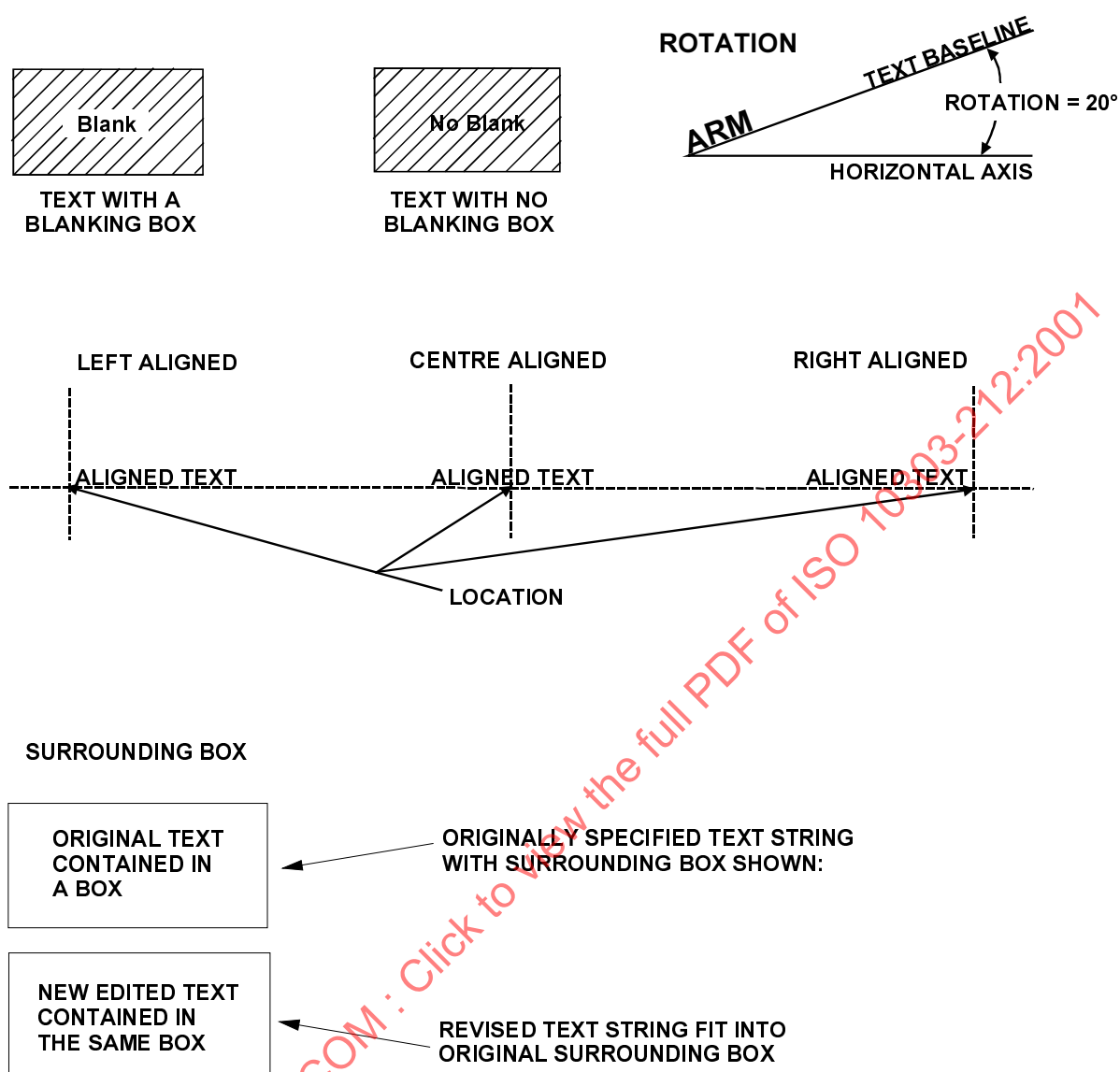


Figure 19 - Text characteristics

#### 4.2.343.1 alignment

The alignment specifies the adjustment of the Text\_string (see 4.2.345) objects contained in the Text with respect to their locations.

The alignment shall contain one of the following text strings:

##### 4.2.343.1.1 centered

centered: The presentation of the Text\_string (see 4.2.345) object is aligned to the middle.

##### 4.2.343.1.2 left

left: The presentation of the Text\_string (see 4.2.345) object is aligned to the left side.

#### 4.2.343.1.3 right

right: The presentation of the Text\_string (see 4.2.345) object is aligned to the right side.

#### 4.2.343.2 blanking\_box

The blanking\_box specifies an area that the text occupies and is used to suppress the visual presentation of all other elements that are within this area.

The blanking\_box need not be specified for a particular Text.

See 4.3.2301 for the application assertion.

#### 4.2.343.3 boundary\_of\_displayed\_box

The boundary\_of\_displayed\_box specifies the border of a rectangular box, composed of annotation curves, that encloses text where one side of the box is parallel to the text baseline.

See 4.3.2300 for the application assertion.

#### 4.2.343.4 default\_appearance

The default\_appearance specifies the preselected appearance of the text when no alternative is specified by the user.

See 4.3.2303 for the application assertion.

#### 4.2.343.5 language\_code

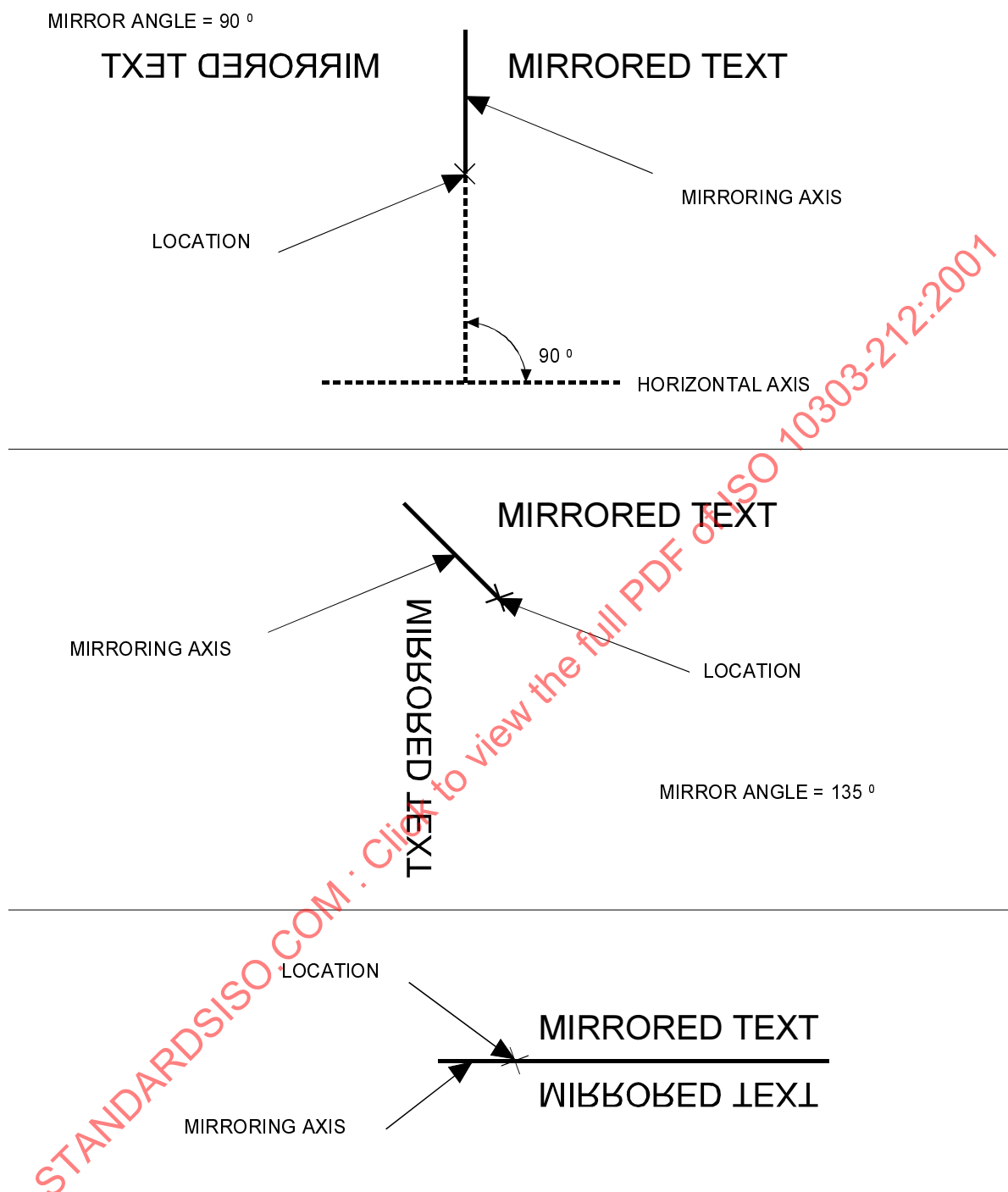
The language\_code specifies a language spoken by human beings to communicate with each other verbally or in written form. The language symbol given in ISO 639 shall be used.

The language\_code need not be specified for a particular Text.

#### 4.2.343.6 mirror\_angle

The mirror\_angle specifies the angle, measured in a counter-clockwise direction from the text baseline, to an axis about which the text is mirrored. The mirror axis and text baseline intersect at the location of the Text.

NOTE See Figure 20 for an illustration of the mirror angle characteristic of text.



**Figure 20 - Mirror angle**

The mirror\_angle need not be specified for a particular Text.

#### 4.2.343.7 surrounding\_box

The surrounding\_box specifies the physical space that the text occupies and is defined by width, the distance of the left-most point of the left-most character to the right-most point of the right-most

## ISO 10303-212:2001(E)

character measured parallel to the text baseline, and height, the distance of the lowest point of the lowest reaching character to the highest point of the highest reaching character measured perpendicular to the text baseline.

The surrounding\_box need not be specified for a particular Text.

See 4.3.2302 for the application assertion.

### 4.2.344 Text\_appearance

A Text\_appearance is a type of Appearance (see 4.2.21) that governs the visual presentation of text.

The data associated with Text\_appearance are the following:

- character\_aspect\_ratio;
- character\_rotation\_angle;
- character\_scale;
- character\_slant\_angle;
- font;
- text\_colour.

NOTE See Figure 21 for an illustration of the characteristics of text appearance.

#### 4.2.344.1 character\_aspect\_ratio

The character\_aspect\_ratio specifies the ratio of the width of the character to the height of the character.

#### 4.2.344.2 character\_rotation\_angle

The character\_rotation\_angle specifies the angular counter-clockwise rotation of each character within the text string in which it appears. The point of rotation is the left-most point of each character at its baseline.

#### 4.2.344.3 character\_scale

The character\_scale specifies the ratio of the size of the text character as defined to the size of the text character as presented.

#### 4.2.344.4 character\_slant\_angle

The character\_slant\_angle specifies the angular distance between vertical aspects of the individual character and an axis perpendicular to the baseline of the character, measured clockwise.



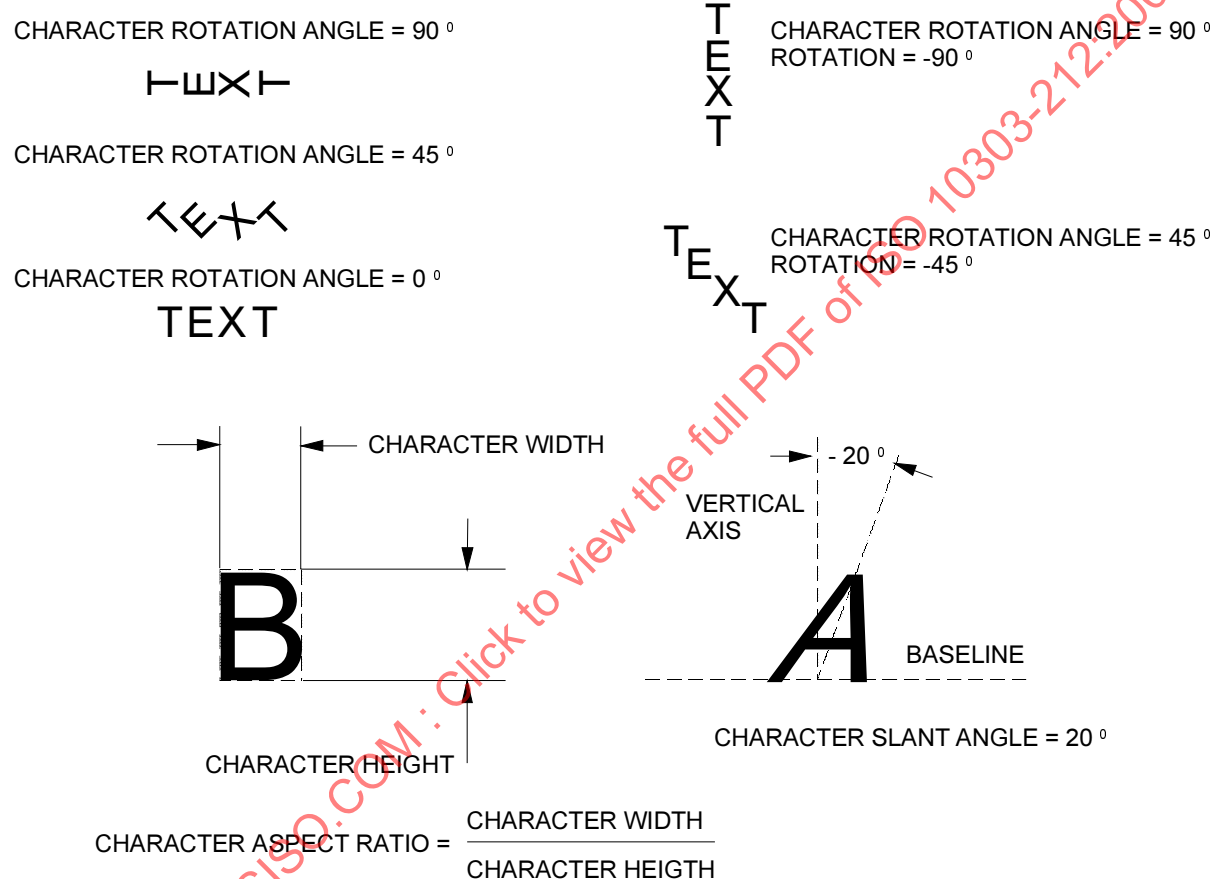
#### 4.2.344.5 font

The font specifies the actual font that is used for the presentation of the text.

See 4.3.2305 for the application assertion.

#### 4.2.344.6 text\_colour

The text\_colour specifies the actual colour that is used for the presentation of the text.



**Figure 21 - Text appearance and rotation of text**

See 4.3.2304 for the application assertion.

#### 4.2.345 Text\_font

A Text\_font is the explicit physical description of each individual character of a character set, including its form and spatial characteristics.

Each Text\_font is either an Externally\_defined\_text\_font (see 4.2.136) or a Predefined\_text\_font (see 4.2.256).

### 4.2.346 Text\_string

A Text\_string is the smallest unit of text and is a collection of one or more characters that convey some human-interpretable information.

The data associated with a Text\_string are the following:

- assigned\_appearance;
- character\_alignment;
- character\_string;
- containing\_text;
- overline\_underline;
- position;
- rotation;
- surrounding\_box.

#### 4.2.346.1 assigned\_appearance

The assigned\_appearance specifies the outlook of the Text\_string object.

The assigned\_appearance need not be specified for a particular Text\_string.

See 4.3.2309 for the application assertion.

#### 4.2.346.2 character\_alignment

The character\_alignment specifies the relative position of successive characters in a string of text.

The value of character\_alignment is one of the following:

- down;
- left;
- right;
- up.

NOTE See 4.2.346.2.1 - 4.2.346.2.4 for the definition of each permissible value for character\_alignment.

##### 4.2.346.2.1 down

down: The presentation of the character is aligned to the bottom line of character body.

**4.2.346.2.2 left**

left: The presentation of the character is aligned to the left side of the character body.

**4.2.346.2.3 right**

right: The presentation of the character is aligned to the right side of the character body.

**4.2.346.2.4 up**

up: The presentation of the character is aligned to the top of the character body.

**4.2.346.3 character\_string**

The `character_string` specifies the list of characters that compose the `Text_string`.

**4.2.346.4 containing\_text**

The `containing_text` specifies the `Text` (see 4.2.342) object containing the `Text_string`.

See 4.3.2308 for the application assertion.

**4.2.346.5 overline\_underline**

The `overline_underline` specifies that there is a line placed either above or beneath the `Text_string`.

The value of `overline_underline` is one of the following:

- `overline`;
- `underline`.

NOTE See 4.2.346.5.1 - 4.2.346.5.2 for the definition of each permissible value for `overline_underline`.

**4.2.346.5.1 overline**

`overline`: The line is placed above the `Text_string`.

**4.2.346.5.2 underline**

`underline`: The line is placed beneath the `Text_string`.

The `overline_underline` need not be specified for a particular `Text_string`.

**4.2.346.6 position**

The `position` specifies the location of the `Text_string` in the placement coordinate system.

See 4.3.2306 for the application assertion.

#### **4.2.346.7 rotation**

The rotation specifies the angle, measured counter-clockwise, between the baseline of the text and the horizontal axis of the coordinate system into which it is being placed.

#### **4.2.346.8 surrounding\_box**

The surrounding\_box specifies the physical space that the text string occupies and is defined by width, the distance of the left-most point of the left-most character to the right-most point of the right-most character measured parallel to the text baseline, and height, the distance of the lowest point of the lowest reaching character to the highest point of the highest reaching character measured perpendicular to the text baseline.

See 4.3.2307 for the application assertion.

#### **4.2.347 Tile**

A Tile is a graphical symbol defined within a containment border and used as the content of a tiling pattern. The containment border defines the edges of the tile. All elements contained within the border are duplicated for each tile. Tiles are placed within a fill area adjacently and do not overlap. The containment border may be blanked.

Each Tile is either an Externally\_defined\_tile (see 4.2.137) or a User\_defined\_tile (see 4.2.356).

The data associated with a Tile are the following:

— overriding\_colour.

##### **4.2.347.1 overriding\_colour**

The overriding\_colour specifies the colour definition that overrides the appearance characteristics already assigned to the elements of the tile.

The overriding\_colour need not be specified for a particular Tile.

See 4.3.2310 for the application assertion.

#### **4.2.348 Typical\_schematic\_node**

A Typical\_schematic\_node is a template for all information that is common to all Schematic\_node (see 4.2.294) objects that use the Typical\_schematic\_node as definition.

The data associated with a `Typical_schematic_node` are the following:

- `consists_of`;
- `coordinate_space`;
- `id`;
- `node_area`.

#### 4.2.348.1 `consists_of`

The `consists_of` specifies the constituents of a `Typical_schematic_node` object.

Each `consists_of` may be one of the following: `Annotation_curve` (see 4.2.14), `Fill_area` (see 4.2.139), or `Text` (see 4.2.342).

See 4.3.2311, 4.3.2314, and 4.3.2315 for the application assertions.

#### 4.2.348.2 `coordinate_space`

The `coordinate_space` specifies the coordinate system that describes the two-dimensional space in which the content of the `Typical_schematic_node` is located.

See 4.3.2312 for the application assertion.

#### 4.2.348.3 `id`

The `id` specifies the identifier of the `Typical_schematic_node`.

#### 4.2.348.4 `node_area`

The `node_area` specifies the zone where the schematic terminal is allowed to be connected with other terminals or connecting lines.

EXAMPLE The zone addressed as 'hot spot' of a connect node.

See 4.3.2313 for the application assertion.

### 4.2.349 `Typical_schematic_text`

A `Typical_schematic_text` is a template for `Schematic_text` (see 4.2.295).

The data associated with a `Typical_schematic_text` are the following:

- `consists_of`;
- `id`.

#### 4.2.349.1 `consists_of`

The `consists_of` specifies the constituents of a `Typical_schematic_text` object.

## ISO 10303-212:2001(E)

See 4.3.2316 for the application assertion.

### 4.2.349.2 id

The id specifies the identifier of the Typical\_schematic\_text.

### 4.2.350 Unstructured\_dimension\_callout

An Unstructured\_dimension\_callout is a type of Dimension\_callout (see 4.2.94) that is a callout wherein a single draughting callout is used and the components of the dimension callout are not semantically identified.

The data associated with an Unstructured\_dimension\_callout are the following:

— basis\_callout.

#### 4.2.350.1 basis\_callout

The basis\_callout specifies the callout that is identified as being unstructured.

See 4.3.2317 for the application assertion.

### 4.2.351 User\_defined\_colour

A User\_defined\_colour is a type of Colour (see 4.2.50) that is defined by an explicit listing of the proportions of blue, green, and red.

The data associated with a User\_defined\_colour are the following:

— blue\_proportion;

— colour\_id;

— green\_proportion;

— red\_proportion.

#### 4.2.351.1 blue\_proportion

The blue\_proportion specifies the level of intensity of the colour blue to be displayed.

#### 4.2.351.2 colour\_id

The colour\_id specifies the identification of a particular user defined colour.

The colour\_id need not be specified for a particular User\_defined\_colour.

#### 4.2.351.3 green\_proportion

The green\_proportion specifies the level of intensity of the colour green to be displayed.

#### 4.2.351.4 red\_proportion

The red\_proportion specifies the level of intensity of the colour red to be displayed.

#### 4.2.352 User\_defined\_data\_element

The User\_defined\_data\_element is a type of Data\_element (see 4.2.70) that is defined in a user specific standard.

NOTE 1 In many cases User\_defined\_data\_element objects are subject of bilateral arrangements among the exchanging enterprises.

NOTE 2 User\_defined\_data\_element objects may be composed from other Data\_element (see 4.2.70) objects by using Data\_element\_relationship (see 4.2.74) objects with relation\_type 'decomposition'.

EXAMPLE 1 A User\_defined\_data\_element object 'Electrical data' comprises all aspects of a drive that are related to the electrical properties of the motor, such as rated voltage, peak voltage, etc.

EXAMPLE 2 A User\_defined\_data\_element object 'Required computer equipment' comprises the data that characterise the hardware and software items needed to run a motor, including the control program, the operating system, type of processor, required capacity of the disk, etc.

The data associated with an User\_defined\_data\_element are the following:

- definition;
- value\_of\_data\_element.

##### 4.2.352.1 definition

The definition specifies a Data\_element\_definition (see 4.2.72) object that specifies the meaning of the associated values.

See 4.3.2318 for the application assertion.

##### 4.2.352.2 value\_of\_data\_element

The value\_of\_data\_element assigns the Data\_element\_value (see 4.2.76) to the User\_defined\_data\_element.

The value\_of\_data\_element need not be specified for a particular User\_defined\_data\_element.

See 4.3.2319 for the application assertion.

#### 4.2.353 User\_defined\_hatching

A User\_defined\_hatching is a type of Fill\_area\_appearance (see 4.2.140) that is defined by an explicit listing of hatch patterns.

## ISO 10303-212:2001(E)

The data associated with an `User_defined_hatching` are the following:

— `defining_pattern`.

### 4.2.353.1 `defining_pattern`

The `defining_pattern` specifies the pattern that serves as the template from which the `User_defined_hatching` is derived.

See 4.3.2320 for the application assertion.

### 4.2.354 `User_defined_line_font`

A `User_defined_line_font` is a type of `Line_font` (see 4.2.189) that is defined by an explicit listing of the visible and invisible segments that make up the pattern of the font.

The data associated with a `User_defined_line_font` are the following:

— `font_id`;

— `pattern`.

#### 4.2.354.1 `font_id`

The `font_id` specifies the identification of a particular line font.

#### 4.2.354.2 `pattern`

The pattern specifies a list of length values of visible and invisible segments.

There shall be two or more pattern for a `User_defined_line_font`.

### 4.2.355 `User_defined_symbol`

A `User_defined_symbol` is a type of `Annotation_symbol` (see 4.2.20) that is a symbol that is defined by an explicit listing of annotation elements that make up the symbol along with their positions within the coordinate system in which the symbol is defined.

The data associated with a `User_defined_symbol` are the following:

— `definition`.

#### 4.2.355.1 `definition`

The definition specifies the `User_defined_symbol_definition` (see 4.2.355) object that serves as a template for the `User_defined_symbol`.

See 4.3.2321 for the application assertion.



### 4.2.356 User\_defined\_symbol\_definition

A `User_defined_symbol_definition` is a collection of annotation elements, along with their placements in a coordinate space, that, taken as a whole, represent a distinct concept.

Each `User_defined_symbol_definition` is either a `Drawing_sheet_layout` (see 4.2.123) or a `Reference_grid_layout` (see 4.2.284).

The data associated with an `User_defined_symbol_definition` are the following:

- `blanking_box`;
- `components`;
- `coordinate_space`;
- `symbol_definition_id`.

#### 4.2.356.1 blanking\_box

The `blanking_box` specifies an area that the symbol occupies and is used to suppress the visual presentation of all other elements that are within this area.

The `blanking_box` need not be specified for a particular `User_defined_symbol_definition`.

See 4.3.2324 for the application assertion.

#### 4.2.356.2 components

The `components` specifies the constituents of a `User_defined_symbol_definition`.

See 4.3.2322 for the application assertion.

#### 4.2.356.3 coordinate\_space

The `coordinate_space` specifies the coordinate system that describes the two-dimensional space in which the constituents of the `User_defined_symbol_definition` are located. The origin of the associated coordinate system is the reference point of the `User_defined_symbol_definition`.

See 4.3.2323 for the application assertion.

#### 4.2.356.4 symbol\_definition\_id

The `symbol_definition_id` specifies the identification of a particular symbol.

### 4.2.357 User\_defined\_tile

A `User_defined_tile` is a type of `Tile` (see 4.2.346) that is defined by an explicit listing of components that make up the tile.

## ISO 10303-212:2001(E)

The data associated with an `User_defined_tile` are the following:

— definition.

### 4.2.357.1 definition

The definition specifies the `User_defined_symbol_definition` (see 4.2.355) object that serves as a template from which the `User_defined_tile` is derived.

See 4.3.2325 for the application assertion.

### 4.2.358 User\_defined\_tiling

A `User_defined_tiling` is a type of `Fill_area_appearance` (see 4.2.140) that is defined by an explicit listing of tiles.

The data associated with a `User_defined_tiling` are the following:

- angle;
- defining\_tile;
- repeat\_vector\_1;
- repeat\_vector\_2;
- scale.

#### 4.2.358.1 angle

The angle specifies the rotation of the tile, measured counter-clockwise, relative to the x-axis of the coordinate system in which the boundary of the fill area is defined.

#### 4.2.358.2 defining\_tile

The `defining_tile` specifies the tile that serves as a template from which the `User_defined_tiling` is derived.

See 4.3.2326 for the application assertion.

#### 4.2.358.3 repeat\_vector\_1

The `repeat_vector_1` specifies the direction and the distance in that direction at which to place the tile relative to the placement of a previous tile.

#### 4.2.358.4 repeat\_vector\_2

The `repeat_vector_2` specifies the secondary direction and the distance in that direction at which to place the tile relative to the placement of a previous tile.

### 4.2.358.5 scale

The scale specifies the ratio between the size of the tile as defined and the size of the tile as presented.

### 4.2.359 Value\_limit

A Value\_limit is a type of Value\_with\_unit (see 4.2.360) that is a qualified numerical value representing either the lower limit or the upper limit of a particular physical characteristic.

EXAMPLE '30.5 max' and '5 min' are examples for a Value\_limit.

The data associated with a Value\_limit are the following:

- limit;
- limit\_qualifier.

#### 4.2.359.1 limit

The limit specifies the value of the limit.

#### 4.2.359.2 limit\_qualifier

The limit\_qualifier specifies the kind of limit.

The following values shall be used:

- maximum;
- minimum.

NOTE See 4.2.359.2.1 - 4.2.359.2.2 for the definition of each predefined value for limit\_qualifier.

##### 4.2.359.2.1 maximum

maximum: The specified limit is an upper limit.

##### 4.2.359.2.2 minimum

minimum: The specified limit is a lower limit.

### 4.2.360 Value\_range

A Value\_range is a type of Value\_with\_unit (see 4.2.360) that is a pair of numerical values representing the range in which the value shall lie.

## ISO 10303-212:2001(E)

The data associated with a Value\_range are the following:

- lower\_limit;
- upper\_limit.

### 4.2.360.1 lower\_limit

The lower\_limit specifies the minimum acceptable value that is constrained by the Value\_range.

### 4.2.360.2 upper\_limit

The upper\_limit specifies the maximum acceptable value that is constrained by the Value\_range.

### 4.2.361 Value\_with\_unit

A Value\_with\_unit is a single numerical measure, or a range of numerical measures with upper, lower, or upper and lower bounds.

Each Value\_with\_unit is either a Numerical\_value (see 4.2.216), a Value\_limit (see 4.2.358), or a Value\_range (see 4.2.359).

The data associated with a Value\_with\_unit are the following:

- significant\_digits;
- unit\_component.

#### 4.2.361.1 significant\_digits

The significant\_digits specifies the number of decimal digits that are relevant for the use of the Value\_with\_unit. If present, the numerical measure or range may be specified using more digits than the significant digits but shall not be specified using less digits.

The significant\_digits need not be specified for a particular Value\_with\_unit.

#### 4.2.361.2 unit\_component

The unit\_component specifies the unit in which the Value\_with\_unit is expressed.

The unit\_component need not be specified for a particular Value\_with\_unit.

### 4.2.362 View\_displayed\_model

A View\_displayed\_model is the identification and assignment of appearance characteristics to a Draughting\_model (see 4.2.118) when it is presented in a Drawing\_view (see 4.2.125).

The data associated with a `View_displayed_model` are the following:

- `clipping`;
- `displayed_model`;
- `overriding_appearance`;
- `presented_in`;
- `scale`;
- `transformation`.

#### 4.2.362.1 clipping

The `clipping` specifies the mathematical information necessary to define a two-dimensional boundary that encloses all viewable geometric and annotation elements of a `Draughting_model` (see 4.2.118). Only those elements, or portions of any elements, that fall within this boundary will be displayed.

See 4.3.2330 for the application assertion.

#### 4.2.362.2 displayed\_model

The `displayed_model` specifies the `Draughting_model` (see 4.2.118) that is presented in the `Drawing_view` (see 4.2.125).

See 4.3.2328 for the application assertion.

#### 4.2.362.3 overriding\_appearance

The `overriding_appearance` specifies the appearance characteristics that are applied to the `Draughting_model` (see 4.2.118) when presented in the `Drawing_view` (see 4.2.125).

The `overriding_appearance` need not be specified for a particular `View_displayed_model`.

See 4.3.2327 for the application assertion.

#### 4.2.362.4 presented\_in

The `presented_in` specifies the `Drawing_view` (see 4.2.125) in which the model will be presented.

See 4.3.2329 for the application assertion.

#### 4.2.362.5 scale

The `scale` specifies the ratio between the size of the elements as defined in the `Draughting_model` (see 4.2.118) and the size of the elements as presented in the `Drawing_view` (see 4.2.125).

#### **4.2.362.6 transformation**

The transformation specifies the mathematical values that define the relationship between elements located in the coordinate system of the Draughting\_model (see 4.2.118) and their location in the coordinate system of the Drawing\_view (see 4.2.125).

#### **4.2.363 View\_placed\_annotation**

A View\_placed\_annotation is a type of Draughting\_annotation (see 4.2.116) that is an annotation that is located in the coordinate system of the drawing view.

The data associated with a View\_placed\_annotation are the following:

- annotation\_layers;
- annotation\_visibility;
- containing\_view.

##### **4.2.363.1 annotation\_layers**

The annotation\_layers specifies the layers that contain the annotation.

See 4.3.2332 for the application assertion.

##### **4.2.363.2 annotation\_visibility**

The annotation\_visibility specifies whether or not each piece of annotation placed within the drawing sheet is visible.

See 4.3.2333 for the application assertion.

##### **4.2.363.3 containing\_view**

The containing\_sheet specifies the drawing\_view in which the View\_placed\_annotation is placed.

See 4.3.2331 for the application assertion.

#### **4.2.364 Visibility**

A Visibility is an indication of whether or not an individual element or collection of elements are displayed in the visual presentation of the drawing. Visibility takes precedence over all other appearance characteristics assigned to the element.

EXAMPLE A construction line is an element not meant for display on a drawing; therefore, visibility would indicate this.

#### **4.2.365 Work\_order**

A Work\_order is the authorization for an Activity (see 4.2.1) to be performed.

The data associated with Work\_order are the following:

- description;
- id;
- is\_controlling
- version\_id;
- work\_order\_type.

#### **4.2.365.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Work\_order.

The description need not be specified for a particular Work\_order.

#### **4.2.365.2 id**

The id specifies the identifier of the Work\_order.

#### **4.2.365.3 is\_controlling**

The is\_controlling specifies the Activity (see 4.2.1) that is controlled by this particular Work\_order.

See 4.3.2334 for the application assertion.

#### **4.2.365.4 version\_id**

The version\_id specifies versioning information for the Work\_order.

The version\_id need not be specified for a particular Work\_order.

#### **4.2.365.5 work\_order\_type**

The work\_order\_type specifies the kind of the Work\_order. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of work\_order\_type is one of the following:

- design deviation permit;
- design release;
- management resolution;
- manufacturing release;
- production deviation permit.

NOTE See 4.2.365.5.1 - 4.2.365.5.5 for the definition of each predefined value for work\_order\_type.

### 4.2.365.5.1 design deviation permit

design deviation permit: An authorization for a deviation from the approved design data.

### 4.2.365.5.2 design release

design release: An authorization for the design of a physical or abstract item and for the creation of a bill of material.

### 4.2.365.5.3 management resolution

management resolution: An authorization by a committee, such as the board of directors, to design or change a product, a product component, or some items.

### 4.2.365.5.4 manufacturing release

manufacturing release: An authorization for the manufacturing process of a product or an item.

### 4.2.365.5.5 production deviation permit

production deviation permit: An authorization for a deviation from the approved manufacturing process.

## 4.2.366 Work\_request

A Work\_request is the solicitation for some type of work to be done.



The data associated with Work\_request are the following:

- description;
- id;
- notified\_person\_or\_organization;
- request\_type;
- requestor;
- scope;
- status;
- version\_id.

#### **4.2.366.1 description**

The description specifies an alphanumeric string containing human-interpretable text that gives further details about the Work\_request.

The description need not be specified for a particular Work\_request.

#### **4.2.366.2 id**

The id specifies the identifier of the Work\_request.

#### **4.2.366.3 notified\_person\_or\_organization**

The notified\_person\_or\_organization specifies the party who shall be informed about the Work\_request and the date when the notification occurred.

See 4.3.2357 for the application assertion.

#### **4.2.366.4 request\_type**

The request\_type specifies the intention of the Work\_request. The value is either user defined or predefined.

## ISO 10303-212:2001(E)

The predefined value of request\_type is one of the following:

- change of standard;
- cost reduction;
- customer rejection;
- customer request;
- durability improvement;
- government regulation;
- procurement alignment;
- production alignment;
- production relief;
- production requirement;
- quality improvement;
- security reason;
- standardization;
- supplier request;
- technical improvement;
- tool improvement.

NOTE See 4.2.366.4.1 - 4.2.366.4.16 for the definition of each predefined value for request\_type.

### 4.2.366.4.1 change of standard

change of standard: A request to translate a standard change into action.

### 4.2.366.4.2 cost reduction

cost reduction: A request aiming at the reduction of engineering and manufacturing costs of an item.

### 4.2.366.4.3 customer rejection

customer rejection: A request resulting from a rejection by a customer.

### 4.2.366.4.4 customer request

customer request: A request resulting from requests by a customer.

### 4.2.366.4.5 durability improvement

durability improvement: A request aiming at a longer life time of a part.

#### **4.2.366.4.6 government regulation**

government regulation: A request resulting from legal requirements.

#### **4.2.366.4.7 procurement alignment**

procurement alignment: A request to adjust the purchasing process of different items.

#### **4.2.366.4.8 production alignment**

production alignment: A request to adjust the manufacturing process of different items.

#### **4.2.366.4.9 production relief**

production relief: A request aiming at a simpler assembly and production process.

#### **4.2.366.4.10 production requirement**

production requirement: A request for an activity necessary from a production point of view.

#### **4.2.366.4.11 quality improvement**

quality improvement: A request aiming at an increased quality of an item.

#### **4.2.366.4.12 security reason**

security reason: A request for an activity necessary from a security point of view.

#### **4.2.366.4.13 standardization**

standardization: A request to unify variants of an item.

#### **4.2.366.4.14 supplier request**

supplier request: A request resulting from requests by a supplier.

#### **4.2.366.4.15 technical improvement**

technical improvement: A request aiming at the technical improvement of an item.

#### **4.2.366.4.16 tool improvement**

tool improvement: A request aiming at a reduction of the wear of tools.

### **4.2.366.5 requestor**

The requestor specifies the Person (see 4.2.237) or Organization (see 4.2.223) who issued the Work\_request and the date when the Work\_request was issued.

The requestor need not be specified for a particular Work\_request.

See 4.3.2358 for the application assertion.

## 4.2.366.6 scope

The scope specifies the items that are affected by the Work\_request.

Each scope may be one of the following: Activity\_method (see 4.2.3), Activity\_relationship (see 4.2.5), Alternate\_item\_relationship (see 4.2.11), Alternate\_item\_relationship (see 4.2.11), Assembly\_component\_relationship (see 4.2.26), Cable\_pull\_information (see 4.2.33), Class\_category\_association (see 4.2.40), Class\_condition\_association (see 4.2.41), Class\_inclusion\_association (see 4.2.42), Class\_specification\_association (see 4.2.44), Class\_structure\_relationship (see 4.2.45), Classification\_system (see 4.2.48), Complex\_product (see 4.2.51), Complex\_product\_relationship (see 4.2.52), Composition\_relationship (see 4.2.55), Configuration (see 4.2.56), Connectivity\_definition (see 4.2.61), Connectivity\_definition\_relationship (see 4.2.62), Data\_element (see 4.2.70), Data\_element\_association (see 4.2.71), Data\_element\_definition (see 4.2.72), Data\_element\_relationship (see 4.2.74), Design\_discipline\_item\_definition (see 4.2.86), Device (see 4.2.88), Device\_relationship (see 4.2.89), Document (see 4.2.101), Document\_file (see 4.2.106), Document\_file\_relationship (see 4.2.107), Document\_representation (see 4.2.110), Document\_version (see 4.2.114), Document\_version\_relationship (see 4.2.115), Drawing (see 4.2.119), Drawing\_sequence (see 4.2.121), Drawing\_sheet (see 4.2.122), Drawing\_sheet\_relationship (see 4.2.124), Function\_definition (see 4.2.145), Function\_definition\_relationship (see 4.2.146), Function\_interface (see 4.2.147), Function\_unit (see 4.2.148), Function\_unit\_relationship (see 4.2.149), Function\_version (see 4.2.150), Function\_version\_relationship (see 4.2.151), Functional\_connectivity\_definition (see 4.2.152), Functional\_connectivity\_definition\_relationship (see 4.2.153), Generic\_note (see 4.2.159), Interface (see 4.2.170), Interface\_port (see 4.2.171), Interface\_terminal (see 4.2.174), Item (see 4.2.178), Item\_definition\_relationship (see 4.2.179), Item\_version (see 4.2.182), Item\_version\_relationship (see 4.2.183), Location (see 4.2.192), Location\_relationship (see 4.2.194), Manufacturing\_configuration (see 4.2.198), Marking (see 4.2.199), Node (see 4.2.208), Node\_relationship (see 4.2.209), Notification (see 4.2.213), Notification\_relationship (see 4.2.214), Path (see 4.2.232), Path\_node (see 4.2.233), Path\_node\_relationship (see 4.2.234), Path\_relationship (see 4.2.235), Physical\_assembly\_relationship (see 4.2.241), Physical\_instance (see 4.2.243), Port (see 4.2.247), Process\_variable (see 4.2.260), Process\_variable\_relationship (see 4.2.261), Product\_class (see 4.2.263), Product\_identification (see 4.2.268), Product\_structure\_relationship (see 4.2.270), Requirement (see 4.2.285), Route (see 4.2.290), Route\_relationship (see 4.2.291), Section (see 4.2.296), Section\_end (see 4.2.297), Section\_interface (see 4.2.298), Section\_interface\_relationship (see 4.2.299), Section\_relationship (see 4.2.300), Signal (see 4.2.309), Signal\_relationship (see 4.2.311), Signal\_value (see 4.2.313), Specification (see 4.2.323), Specification\_category (see 4.2.324), Specification\_expression (see 4.2.326), Specification\_inclusion (see 4.2.327), Technical\_system (see 4.2.336), Technical\_system\_relationship (see 4.2.337), or Terminal (see 4.2.338).

See 4.3.2335, 4.3.2336, 4.3.2337, 4.3.2338, 4.3.2339, 4.3.2340, 4.3.2341, 4.3.2342, 4.3.2343, 4.3.2344, 4.3.2345, 4.3.2346, 4.3.2347, 4.3.2348, 4.3.2349, 4.3.2350, 4.3.2351, 4.3.2352, 4.3.2353, 4.3.2354, 4.3.2355, 4.3.2356, 4.3.2359, 4.3.2360, 4.3.2361, 4.3.2362, 4.3.2363, 4.3.2364, 4.3.2365, 4.3.2366, 4.3.2367, 4.3.2368, 4.3.2369, 4.3.2370, 4.3.2371, 4.3.2372, 4.3.2373, 4.3.2374, 4.3.2375, 4.3.2376, 4.3.2377, 4.3.2378, 4.3.2379, 4.3.2380, 4.3.2381, 4.3.2382, 4.3.2383, 4.3.2384, 4.3.2385, 4.3.2386, 4.3.2387, 4.3.2388, 4.3.2389, 4.3.2390, 4.3.2391, 4.3.2392, 4.3.2393, 4.3.2394, 4.3.2395, 4.3.2396, 4.3.2397, 4.3.2398, 4.3.2399, 4.3.2400, 4.3.2401, 4.3.2402, 4.3.2403, 4.3.2404, 4.3.2405, 4.3.2406, 4.3.2407, 4.3.2408, 4.3.2409, 4.3.2410, 4.3.2411, 4.3.2412, 4.3.2413, 4.3.2414, 4.3.2415,

4.3.2416, 4.3.2417, 4.3.2418, 4.3.2419, 4.3.2420, 4.3.2421, 4.3.2422, 4.3.2423, 4.3.2424, 4.3.2425, and 4.3.2426 for the application assertions.

#### **4.2.366.7 status**

The status specifies the position of affairs for the Work\_request.

EXAMPLE The status information 'ongoing' would be an example of status.

NOTE The values and sequence of status are company specific.

#### **4.2.366.8 version\_id**

The version\_id specifies versioning information for the Work\_request.

The version\_id need not be specified for a particular Work\_request.

### **4.3 Application assertions**

This subclause specifies the application assertions for the Core Data for electrical design and installation application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

#### **4.3.1 Activity to Activity\_method**

Each Activity refers to zero or one Activity\_method in the role of chosen\_method. Each Activity\_method acts as chosen\_method for zero, one, or more Activity objects.

#### **4.3.2 Activity to Date\_and\_person\_or\_organization**

Each Activity refers to zero or one Date\_and\_person\_or\_organization in the role of requestor. Each Date\_and\_person\_or\_organization acts as requestor for zero, one, or more Activity objects.

#### **4.3.3 Activity to Date\_time**

Each Activity refers to zero or one Date\_time in the role of actual\_end\_date. Each Date\_time acts as actual\_end\_date for zero, one, or more Activity objects.

#### **4.3.4 Activity to Date\_time**

Each Activity refers to zero or one Date\_time in the role of actual\_start\_date. Each Date\_time acts as actual\_start\_date for zero, one, or more Activity objects.

#### **4.3.1 Activity to Date\_time**

Each Activity refers to zero or one Date\_time in the role of planned\_end\_date. Each Date\_time acts as planned\_end\_date for zero, one, or more Activity objects.

#### **4.3.2 Activity to Date\_time**

Each Activity refers to zero or one Date\_time in the role of planned\_start\_date. Each Date\_time acts as planned\_start\_date for zero, one, or more Activity objects.

#### **4.3.3 Activity to Duration**

Each Activity refers to zero or one Duration in the role of planned\_end\_date. Each Duration acts as planned\_end\_date for zero, one, or more Activity objects.

#### **4.3.4 Activity to Event\_reference**

Each Activity refers to zero or one Event\_reference in the role of planned\_end\_date. Each Event\_reference acts as planned\_end\_date for zero, one, or more Activity objects.

#### **4.3.5 Activity to Event\_reference**

Each Activity refers to zero or one Event\_reference in the role of planned\_start\_date. Each Event\_reference acts as planned\_start\_date for zero, one, or more Activity objects.

#### **4.3.6 Activity to Organization**

Each Activity refers to zero, one, or more Organization objects in the role of concerned\_organization. Each Organization acts as concerned\_organization for zero, one, or more Activity objects.

#### **4.3.7 Activity to Organization**

Each Activity refers to zero, one, or more Organization objects in the role of supplying\_organization. Each Organization acts as supplying\_organization for zero, one, or more Activity objects.

#### **4.3.8 Activity to Work\_request**

Each Activity refers to zero, one, or more Work\_request objects in the role of resolved\_request. Each Work\_request acts as resolved\_request for zero, one, or more Activity objects.

#### **4.3.9 Activity\_element to Activity**

Each Activity\_element refers to exactly one Activity in the role of associated\_activity. Each Activity acts as associated\_activity for zero, one, or more Activity\_element objects.

#### **4.3.10 Activity\_element to Activity\_method**

Each Activity\_element refers to exactly one Activity\_method in the role of element. Each Activity\_method acts as element for zero, one, or more Activity\_element objects.

#### **4.3.11 Activity\_element to Activity\_relationship**

Each Activity\_element refers to exactly one Activity\_relationship in the role of element. Each Activity\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.12 Activity\_element to Alternate\_item\_relationship**

Each Activity\_element refers to exactly one Alternate\_item\_relationship in the role of element. Each Alternate\_item\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.13 Activity\_element to Alternate\_item\_relationship**

Each Activity\_element refers to exactly one Alternate\_item\_relationship in the role of element. Each Alternate\_item\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.14 Activity\_element to Assembly\_component\_relationship**

Each Activity\_element refers to exactly one Assembly\_component\_relationship in the role of element. Each Assembly\_component\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.15 Activity\_element to Cable\_pull\_information**

Each Activity\_element refers to exactly one Cable\_pull\_information in the role of element. Each Cable\_pull\_information acts as element for zero, one, or more Activity\_element objects.

#### **4.3.16 Activity\_element to Class\_category\_association**

Each Activity\_element refers to exactly one Class\_category\_association in the role of element. Each Class\_category\_association acts as element for zero, one, or more Activity\_element objects.

#### **4.3.17 Activity\_element to Class\_condition\_association**

Each Activity\_element refers to exactly one Class\_condition\_association in the role of element. Each Class\_condition\_association acts as element for zero, one, or more Activity\_element objects.

#### **4.3.18 Activity\_element to Class\_inclusion\_association**

Each Activity\_element refers to exactly one Class\_inclusion\_association in the role of element. Each Class\_inclusion\_association acts as element for zero, one, or more Activity\_element objects.

#### **4.3.19 Activity\_element to Class\_specification\_association**

Each Activity\_element refers to exactly one Class\_specification\_association in the role of element. Each Class\_specification\_association acts as element for zero, one, or more Activity\_element objects.

#### **4.3.20 Activity\_element to Class\_structure\_relationship**

Each Activity\_element refers to exactly one Class\_structure\_relationship in the role of element. Each Class\_structure\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.21 Activity\_element to Classification\_system**

Each Activity\_element refers to exactly one Classification\_system in the role of element. Each Classification\_system acts as element for zero, one, or more Activity\_element objects.

#### **4.3.22 Activity\_element to Complex\_product**

Each Activity\_element refers to exactly one Complex\_product in the role of element. Each Complex\_product acts as element for zero, one, or more Activity\_element objects.

#### **4.3.23 Activity\_element to Complex\_product\_relationship**

Each Activity\_element refers to exactly one Complex\_product\_relationship in the role of element. Each Complex\_product\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.24 Activity\_element to Composition\_relationship**

Each Activity\_element refers to exactly one Composition\_relationship in the role of element. Each Composition\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.25 Activity\_element to Configuration**

Each Activity\_element refers to exactly one Configuration in the role of element. Each Configuration acts as element for zero, one, or more Activity\_element objects.

#### **4.3.26 Activity\_element to Connectivity\_definition**

Each Activity\_element refers to exactly one Connectivity\_definition in the role of element. Each Connectivity\_definition acts as element for zero, one, or more Activity\_element objects.

#### **4.3.27 Activity\_element to Connectivity\_definition\_relationship**

Each Activity\_element refers to exactly one Connectivity\_definition\_relationship in the role of element. Each Connectivity\_definition\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.28 Activity\_element to Data\_element**

Each Activity\_element refers to exactly one Data\_element in the role of element. Each Data\_element acts as element for zero, one, or more Activity\_element objects.

#### **4.3.29 Activity\_element to Data\_element\_association**

Each Activity\_element refers to exactly one Data\_element\_association in the role of element. Each Data\_element\_association acts as element for zero, one, or more Activity\_element objects.

#### **4.3.30 Activity\_element to Data\_element\_definition**

Each Activity\_element refers to exactly one Data\_element\_definition in the role of element. Each Data\_element\_definition acts as element for zero, one, or more Activity\_element objects.



#### **4.3.31 Activity\_element to Data\_element\_relationship**

Each Activity\_element refers to exactly one Data\_element\_relationship in the role of element. Each Data\_element\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.32 Activity\_element to Design\_discipline\_item\_definition**

Each Activity\_element refers to exactly one Design\_discipline\_item\_definition in the role of element. Each Design\_discipline\_item\_definition acts as element for zero, one, or more Activity\_element objects.

#### **4.3.33 Activity\_element to Device**

Each Activity\_element refers to exactly one Device in the role of element. Each Device acts as element for zero, one, or more Activity\_element objects.

#### **4.3.34 Activity\_element to Device\_relationship**

Each Activity\_element refers to exactly one Device\_relationship in the role of element. Each Device\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.35 Activity\_element to Document**

Each Activity\_element refers to exactly one Document in the role of element. Each Document acts as element for zero, one, or more Activity\_element objects.

#### **4.3.36 Activity\_element to Document\_file**

Each Activity\_element refers to exactly one Document\_file in the role of element. Each Document\_file acts as element for zero, one, or more Activity\_element objects.

#### **4.3.37 Activity\_element to Document\_file\_relationship**

Each Activity\_element refers to exactly one Document\_file\_relationship in the role of element. Each Document\_file\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.38 Activity\_element to Document\_representation**

Each Activity\_element refers to exactly one Document\_representation in the role of element. Each Document\_representation acts as element for zero, one, or more Activity\_element objects.

#### **4.3.39 Activity\_element to Document\_version**

Each Activity\_element refers to exactly one Document\_version in the role of element. Each Document\_version acts as element for zero, one, or more Activity\_element objects.

#### **4.3.40 Activity\_element to Document\_version\_relationship**

Each Activity\_element refers to exactly one Document\_version\_relationship in the role of element. Each Document\_version\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.41 Activity\_element to Drawing**

Each Activity\_element refers to exactly one Drawing in the role of element. Each Drawing acts as element for zero, one, or more Activity\_element objects.

#### **4.3.42 Activity\_element to Drawing\_sequence**

Each Activity\_element refers to exactly one Drawing\_sequence in the role of element. Each Drawing\_sequence acts as element for zero, one, or more Activity\_element objects.

#### **4.3.43 Activity\_element to Drawing\_sheet**

Each Activity\_element refers to exactly one Drawing\_sheet in the role of element. Each Drawing\_sheet acts as element for zero, one, or more Activity\_element objects.

#### **4.3.44 Activity\_element to Drawing\_sheet\_relationship**

Each Activity\_element refers to exactly one Drawing\_sheet\_relationship in the role of element. Each Drawing\_sheet\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.45 Activity\_element to Function\_definition**

Each Activity\_element refers to exactly one Function\_definition in the role of element. Each Function\_definition acts as element for zero, one, or more Activity\_element objects.

#### **4.3.46 Activity\_element to Function\_definition\_relationship**

Each Activity\_element refers to exactly one Function\_definition\_relationship in the role of element. Each Function\_definition\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.47 Activity\_element to Function\_interface**

Each Activity\_element refers to exactly one Function\_interface in the role of element. Each Function\_interface acts as element for zero, one, or more Activity\_element objects.

#### **4.3.48 Activity\_element to Function\_unit**

Each Activity\_element refers to exactly one Function\_unit in the role of element. Each Function\_unit acts as element for zero, one, or more Activity\_element objects.

#### **4.3.49 Activity\_element to Function\_unit\_relationship**

Each Activity\_element refers to exactly one Function\_unit\_relationship in the role of element. Each Function\_unit\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.50 Activity\_element to Function\_version**

Each Activity\_element refers to exactly one Function\_version in the role of element. Each Function\_version acts as element for zero, one, or more Activity\_element objects.

#### **4.3.51 Activity\_element to Function\_version\_relationship**

Each Activity\_element refers to exactly one Function\_version\_relationship in the role of element. Each Function\_version\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.52 Activity\_element to Functional\_connectivity\_definition**

Each Activity\_element refers to exactly one Functional\_connectivity\_definition in the role of element. Each Functional\_connectivity\_definition acts as element for zero, one, or more Activity\_element objects.

#### **4.3.53 Activity\_element to Functional\_connectivity\_definition\_relationship**

Each Activity\_element refers to exactly one Functional\_connectivity\_definition\_relationship in the role of element. Each Functional\_connectivity\_definition\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.54 Activity\_element to Generic\_note**

Each Activity\_element refers to exactly one Generic\_note in the role of element. Each Generic\_note acts as element for zero, one, or more Activity\_element objects.

#### **4.3.55 Activity\_element to Interface**

Each Activity\_element refers to exactly one Interface in the role of element. Each Interface acts as element for zero, one, or more Activity\_element objects.

#### **4.3.56 Activity\_element to Interface\_port**

Each Activity\_element refers to exactly one Interface\_port in the role of element. Each Interface\_port acts as element for zero, one, or more Activity\_element objects.

#### **4.3.57 Activity\_element to Interface\_terminal**

Each Activity\_element refers to exactly one Interface\_terminal in the role of element. Each Interface\_terminal acts as element for zero, one, or more Activity\_element objects.

#### **4.3.58 Activity\_element to Item**

Each Activity\_element refers to exactly one Item in the role of element. Each Item acts as element for zero, one, or more Activity\_element objects.

#### **4.3.59 Activity\_element to Item\_definition\_relationship**

Each Activity\_element refers to exactly one Item\_definition\_relationship in the role of element. Each Item\_definition\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.60 Activity\_element to Item\_version**

Each Activity\_element refers to exactly one Item\_version in the role of element. Each Item\_version acts as element for zero, one, or more Activity\_element objects.

#### **4.3.61 Activity\_element to Item\_version\_relationship**

Each Activity\_element refers to exactly one Item\_version\_relationship in the role of element. Each Item\_version\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.62 Activity\_element to Location**

Each Activity\_element refers to exactly one Location in the role of element. Each Location acts as element for zero, one, or more Activity\_element objects.

#### **4.3.63 Activity\_element to Location\_relationship**

Each Activity\_element refers to exactly one Location\_relationship in the role of element. Each Location\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.64 Activity\_element to Manufacturing\_configuration**

Each Activity\_element refers to exactly one Manufacturing\_configuration in the role of element. Each Manufacturing\_configuration acts as element for zero, one, or more Activity\_element objects.

#### **4.3.65 Activity\_element to Marking**

Each Activity\_element refers to exactly one Marking in the role of element. Each Marking acts as element for zero, one, or more Activity\_element objects.

#### **4.3.66 Activity\_element to Node**

Each Activity\_element refers to exactly one Node in the role of element. Each Node acts as element for zero, one, or more Activity\_element objects.

#### **4.3.67 Activity\_element to Node\_relationship**

Each Activity\_element refers to exactly one Node\_relationship in the role of element. Each Node\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.68 Activity\_element to Notification**

Each Activity\_element refers to exactly one Notification in the role of element. Each Notification acts as element for zero, one, or more Activity\_element objects.

#### **4.3.69 Activity\_element to Notification\_relationship**

Each Activity\_element refers to exactly one Notification\_relationship in the role of element. Each Notification\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.70 Activity\_element to Path**

Each Activity\_element refers to exactly one Path in the role of element. Each Path acts as element for zero, one, or more Activity\_element objects.

#### **4.3.71 Activity\_element to Path\_node**

Each Activity\_element refers to exactly one Path\_node in the role of element. Each Path\_node acts as element for zero, one, or more Activity\_element objects.

#### **4.3.72 Activity\_element to Path\_node\_relationship**

Each Activity\_element refers to exactly one Path\_node\_relationship in the role of element. Each Path\_node\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.73 Activity\_element to Path\_relationship**

Each Activity\_element refers to exactly one Path\_relationship in the role of element. Each Path\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.74 Activity\_element to Physical\_assembly\_relationship**

Each Activity\_element refers to exactly one Physical\_assembly\_relationship in the role of element. Each Physical\_assembly\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.75 Activity\_element to Physical\_instance**

Each Activity\_element refers to exactly one Physical\_instance in the role of element. Each Physical\_instance acts as element for zero, one, or more Activity\_element objects.

#### **4.3.76 Activity\_element to Port**

Each Activity\_element refers to exactly one Port in the role of element. Each Port acts as element for zero, one, or more Activity\_element objects.

#### **4.3.77 Activity\_element to Process\_variable**

Each Activity\_element refers to exactly one Process\_variable in the role of element. Each Process\_variable acts as element for zero, one, or more Activity\_element objects.

#### **4.3.78 Activity\_element to Process\_variable\_relationship**

Each Activity\_element refers to exactly one Process\_variable\_relationship in the role of element. Each Process\_variable\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.79 Activity\_element to Product\_class**

Each Activity\_element refers to exactly one Product\_class in the role of element. Each Product\_class acts as element for zero, one, or more Activity\_element objects.

#### **4.3.80 Activity\_element to Product\_identification**

Each Activity\_element refers to exactly one Product\_identification in the role of element. Each Product\_identification acts as element for zero, one, or more Activity\_element objects.

#### **4.3.81 Activity\_element to Product\_structure\_relationship**

Each Activity\_element refers to exactly one Product\_structure\_relationship in the role of element. Each Product\_structure\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.82 Activity\_element to Requirement**

Each Activity\_element refers to exactly one Requirement in the role of element. Each Requirement acts as element for zero, one, or more Activity\_element objects.

#### **4.3.83 Activity\_element to Route**

Each Activity\_element refers to exactly one Route in the role of element. Each Route acts as element for zero, one, or more Activity\_element objects.

#### **4.3.84 Activity\_element to Route\_relationship**

Each Activity\_element refers to exactly one Route\_relationship in the role of element. Each Route\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.85 Activity\_element to Section**

Each Activity\_element refers to exactly one Section in the role of element. Each Section acts as element for zero, one, or more Activity\_element objects.

#### **4.3.86 Activity\_element to Section\_end**

Each Activity\_element refers to exactly one Section\_end in the role of element. Each Section\_end acts as element for zero, one, or more Activity\_element objects.

#### **4.3.87 Activity\_element to Section\_interface**

Each Activity\_element refers to exactly one Section\_interface in the role of element. Each Section\_interface acts as element for zero, one, or more Activity\_element objects.

**4.3.88 Activity\_element to Section\_interface\_relationship**

Each Activity\_element refers to exactly one Section\_interface\_relationship in the role of element. Each Section\_interface\_relationship acts as element for zero, one, or more Activity\_element objects.

**4.3.89 Activity\_element to Section\_relationship**

Each Activity\_element refers to exactly one Section\_relationship in the role of element. Each Section\_relationship acts as element for zero, one, or more Activity\_element objects.

**4.3.90 Activity\_element to Signal**

Each Activity\_element refers to exactly one Signal in the role of element. Each Signal acts as element for zero, one, or more Activity\_element objects.

**4.3.91 Activity\_element to Signal\_relationship**

Each Activity\_element refers to exactly one Signal\_relationship in the role of element. Each Signal\_relationship acts as element for zero, one, or more Activity\_element objects.

**4.3.92 Activity\_element to Signal\_value**

Each Activity\_element refers to exactly one Signal\_value in the role of element. Each Signal\_value acts as element for zero, one, or more Activity\_element objects.

**4.3.93 Activity\_element to Specification**

Each Activity\_element refers to exactly one Specification in the role of element. Each Specification acts as element for zero, one, or more Activity\_element objects.

**4.3.94 Activity\_element to Specification\_category**

Each Activity\_element refers to exactly one Specification\_category in the role of element. Each Specification\_category acts as element for zero, one, or more Activity\_element objects.

**4.3.95 Activity\_element to Specification\_expression**

Each Activity\_element refers to exactly one Specification\_expression in the role of element. Each Specification\_expression acts as element for zero, one, or more Activity\_element objects.

**4.3.96 Activity\_element to Specification\_inclusion**

Each Activity\_element refers to exactly one Specification\_inclusion in the role of element. Each Specification\_inclusion acts as element for zero, one, or more Activity\_element objects.

**4.3.97 Activity\_element to Technical\_system**

Each Activity\_element refers to exactly one Technical\_system in the role of element. Each Technical\_system acts as element for zero, one, or more Activity\_element objects.

#### **4.3.98 Activity\_element to Technical\_system\_relationship**

Each Activity\_element refers to exactly one Technical\_system\_relationship in the role of element. Each Technical\_system\_relationship acts as element for zero, one, or more Activity\_element objects.

#### **4.3.99 Activity\_element to Terminal**

Each Activity\_element refers to exactly one Terminal in the role of element. Each Terminal acts as element for zero, one, or more Activity\_element objects.

#### **4.3.100 Activity\_method\_assignment to Activity\_method**

Each Activity\_method\_assignment refers to exactly one Activity\_method in the role of assigned\_method. Each Activity\_method acts as assigned\_method for zero, one, or more Activity\_method\_assignment objects.

#### **4.3.101 Activity\_method\_assignment to Work\_request**

Each Activity\_method\_assignment refers to exactly one Work\_request in the role of assigned\_work\_request. Each Work\_request acts as assigned\_work\_request for zero, one, or more Activity\_method\_assignment objects.

#### **4.3.102 Activity\_relationship to Activity**

Each Activity\_relationship refers to exactly one Activity in the role of related. Each Activity acts as related for zero, one, or more Activity\_relationship objects.

#### **4.3.103 Activity\_relationship to Activity**

Each Activity\_relationship refers to exactly one Activity in the role of relating. Each Activity acts as relating for zero, one, or more Activity\_relationship objects.

#### **4.3.104 Aggregated\_value to Data\_element\_value**

Each Aggregated\_value refers to one or more Data\_element\_value objects in the role of member\_definition. Each Data\_element\_value acts as member\_definition for zero, one, or more Aggregated\_value objects.

#### **4.3.105 Alias\_designation to Device**

Each Alias\_designation is\_applied\_to exactly one Device. Each Device is related to zero, one, or more Alias\_designation objects.

#### **4.3.106 Alias\_designation to Document\_representation**

Each Alias\_designation is\_applied\_to exactly one Document\_representation. Each Document\_representation is related to zero, one, or more Alias\_designation objects.



#### **4.3.107 Alias\_designation to Drawing**

Each Alias\_designation is\_applied\_to exactly one Drawing. Each Drawing is related to zero, one, or more Alias\_designation objects.

#### **4.3.108 Alias\_designation to Drawing\_sheet**

Each Alias\_designation is\_applied\_to exactly one Drawing\_sheet. Each Drawing\_sheet is related to zero, one, or more Alias\_designation objects.

#### **4.3.109 Alias\_designation to Function\_unit**

Each Alias\_designation is\_applied\_to exactly one Function\_unit. Each Function\_unit is related to zero, one, or more Alias\_designation objects.

#### **4.3.110 Alias\_designation to Location**

Each Alias\_designation is\_applied\_to exactly one Location. Each Location is related to zero, one, or more Alias\_designation objects.

#### **4.3.111 Alias\_designation to Object\_designation**

Each Alias\_designation refers to exactly one Object\_designation in the role of alias\_extended\_designation. Each Object\_designation acts as alias\_extended\_designation for zero, one, or more Alias\_designation objects.

#### **4.3.112 Alias\_designation to Organization**

Each Alias\_designation refers to zero or one Organization in the role of alias\_scope. Each Organization acts as alias\_scope for zero, one, or more Alias\_designation objects.

#### **4.3.113 Alias\_designation to Port**

Each Alias\_designation is\_applied\_to exactly one Port. Each Port is related to zero, one, or more Alias\_designation objects.

#### **4.3.114 Alias\_designation to Product\_component**

Each Alias\_designation is\_applied\_to exactly one Product\_component. Each Product\_component is related to zero, one, or more Alias\_designation objects.

#### **4.3.115 Alias\_designation to Signal**

Each Alias\_designation is\_applied\_to exactly one Signal. Each Signal is related to zero, one, or more Alias\_designation objects.

#### **4.3.116 Alias\_designation to Technical\_system**

Each Alias\_designation is\_applied\_to exactly one Technical\_system. Each Technical\_system is related to zero, one, or more Alias\_designation objects.

#### **4.3.117 Alias\_designation to Terminal**

Each Alias\_designation is\_applied\_to exactly one Terminal. Each Terminal is related to zero, one, or more Alias\_designation objects.

#### **4.3.118 Alias\_identification to Approval\_status**

Each Alias\_identification is\_applied\_to exactly one Approval\_status. Each Approval\_status is related to zero, one, or more Alias\_identification objects.

#### **4.3.119 Alias\_identification to Classification\_attribute**

Each Alias\_identification is\_applied\_to exactly one Classification\_attribute. Each Classification\_attribute is related to zero, one, or more Alias\_identification objects.

#### **4.3.120 Alias\_identification to Classification\_system**

Each Alias\_identification is\_applied\_to exactly one Classification\_system. Each Classification\_system is related to zero, one, or more Alias\_identification objects.

#### **4.3.121 Alias\_identification to Complex\_product**

Each Alias\_identification is\_applied\_to exactly one Complex\_product. Each Complex\_product is related to zero, one, or more Alias\_identification objects.

#### **4.3.122 Alias\_identification to Component\_colour**

Each Alias\_identification is\_applied\_to exactly one Component\_colour. Each Component\_colour is related to zero, one, or more Alias\_identification objects.

#### **4.3.123 Alias\_identification to Connectivity\_definition**

Each Alias\_identification is\_applied\_to exactly one Connectivity\_definition. Each Connectivity\_definition is related to zero, one, or more Alias\_identification objects.

#### **4.3.124 Alias\_identification to Data\_element\_definition**

Each Alias\_identification is\_applied\_to exactly one Data\_element\_definition. Each Data\_element\_definition is related to zero, one, or more Alias\_identification objects.

#### **4.3.125 Alias\_identification to Design\_discipline\_item\_definition**

Each Alias\_identification is\_applied\_to exactly one Design\_discipline\_item\_definition. Each Design\_discipline\_item\_definition is related to zero, one, or more Alias\_identification objects.

#### **4.3.126 Alias\_identification to Device**

Each Alias\_identification is\_applied\_to exactly one Device. Each Device is related to zero, one, or more Alias\_identification objects.

#### **4.3.127 Alias\_identification to Document**

Each Alias\_identification is\_applied\_to exactly one Document. Each Document is related to zero, one, or more Alias\_identification objects.

#### **4.3.128 Alias\_identification to Document\_representation**

Each Alias\_identification is\_applied\_to exactly one Document\_representation. Each Document\_representation is related to zero, one, or more Alias\_identification objects.

#### **4.3.129 Alias\_identification to Document\_type\_property**

Each Alias\_identification is\_applied\_to exactly one Document\_type\_property. Each Document\_type\_property is related to zero, one, or more Alias\_identification objects.

#### **4.3.130 Alias\_identification to Document\_version**

Each Alias\_identification is\_applied\_to exactly one Document\_version. Each Document\_version is related to zero, one, or more Alias\_identification objects.

#### **4.3.131 Alias\_identification to Drawing**

Each Alias\_identification is\_applied\_to exactly one Drawing. Each Drawing is related to zero, one, or more Alias\_identification objects.

#### **4.3.132 Alias\_identification to Drawing\_sheet**

Each Alias\_identification is\_applied\_to exactly one Drawing\_sheet. Each Drawing\_sheet is related to zero, one, or more Alias\_identification objects.

#### **4.3.133 Alias\_identification to Function\_definition**

Each Alias\_identification is\_applied\_to exactly one Function\_definition. Each Function\_definition is related to zero, one, or more Alias\_identification objects.

#### **4.3.134 Alias\_identification to Function\_unit**

Each Alias\_identification is\_applied\_to exactly one Function\_unit. Each Function\_unit is related to zero, one, or more Alias\_identification objects.

#### **4.3.135 Alias\_identification to Functional\_connectivity\_definition**

Each Alias\_identification is\_applied\_to exactly one Functional\_connectivity\_definition. Each Functional\_connectivity\_definition is related to zero, one, or more Alias\_identification objects.

#### **4.3.136 Alias\_identification to Functionality**

Each Alias\_identification is\_applied\_to exactly one Functionality. Each Functionality is related to zero, one, or more Alias\_identification objects.

#### **4.3.137 Alias\_identification to General\_classification**

Each Alias\_identification is\_applied\_to exactly one General\_classification. Each General\_classification is related to zero, one, or more Alias\_identification objects.

#### **4.3.138 Alias\_identification to Interface\_port**

Each Alias\_identification is\_applied\_to exactly one Interface\_port. Each Interface\_port is related to zero, one, or more Alias\_identification objects.

#### **4.3.139 Alias\_identification to Interface\_terminal**

Each Alias\_identification is\_applied\_to exactly one Interface\_terminal. Each Interface\_terminal is related to zero, one, or more Alias\_identification objects.

#### **4.3.140 Alias\_identification to Item**

Each Alias\_identification is\_applied\_to exactly one Item. Each Item is related to zero, one, or more Alias\_identification objects.

#### **4.3.141 Alias\_identification to Item\_version**

Each Alias\_identification is\_applied\_to exactly one Item\_version. Each Item\_version is related to zero, one, or more Alias\_identification objects.

#### **4.3.142 Alias\_identification to Location**

Each Alias\_identification is\_applied\_to exactly one Location. Each Location is related to zero, one, or more Alias\_identification objects.

#### **4.3.143 Alias\_identification to Node**

Each Alias\_identification is\_applied\_to exactly one Node. Each Node is related to zero, one, or more Alias\_identification objects.

#### **4.3.144 Alias\_identification to Notification**

Each Alias\_identification is\_applied\_to exactly one Notification. Each Notification is related to zero, one, or more Alias\_identification objects.

#### **4.3.145 Alias\_identification to Organization**

Each Alias\_identification refers to zero or one Organization in the role of alias\_scope. Each Organization acts as alias\_scope for zero, one, or more Alias\_identification objects.

#### **4.3.146 Alias\_identification to Organization**

Each Alias\_identification is\_applied\_to exactly one Organization. Each Organization is related to zero, one, or more Alias\_identification objects.

#### **4.3.147 Alias\_identification to Path**

Each Alias\_identification is\_applied\_to exactly one Path. Each Path is related to zero, one, or more Alias\_identification objects.

#### **4.3.148 Alias\_identification to Path\_node**

Each Alias\_identification is\_applied\_to exactly one Path\_node. Each Path\_node is related to zero, one, or more Alias\_identification objects.

#### **4.3.149 Alias\_identification to Physical\_instance**

Each Alias\_identification is\_applied\_to exactly one Physical\_instance. Each Physical\_instance is related to zero, one, or more Alias\_identification objects.

#### **4.3.150 Alias\_identification to Port**

Each Alias\_identification is\_applied\_to exactly one Port. Each Port is related to zero, one, or more Alias\_identification objects.

#### **4.3.151 Alias\_identification to Process\_variable**

Each Alias\_identification is\_applied\_to exactly one Process\_variable. Each Process\_variable is related to zero, one, or more Alias\_identification objects.

#### **4.3.152 Alias\_identification to Product\_class**

Each Alias\_identification is\_applied\_to exactly one Product\_class. Each Product\_class is related to zero, one, or more Alias\_identification objects.

#### **4.3.153 Alias\_identification to Product\_identification**

Each Alias\_identification is\_applied\_to exactly one Product\_identification. Each Product\_identification is related to zero, one, or more Alias\_identification objects.

#### **4.3.154 Alias\_identification to Requirement**

Each Alias\_identification is\_applied\_to exactly one Requirement. Each Requirement is related to zero, one, or more Alias\_identification objects.

#### **4.3.155 Alias\_identification to Route**

Each Alias\_identification is\_applied\_to exactly one Route. Each Route is related to zero, one, or more Alias\_identification objects.

#### **4.3.156 Alias\_identification to Section**

Each Alias\_identification is\_applied\_to exactly one Section. Each Section is related to zero, one, or more Alias\_identification objects.

#### **4.3.157 Alias\_identification to Section\_interface**

Each Alias\_identification is\_applied\_to exactly one Section\_interface. Each Section\_interface is related to zero, one, or more Alias\_identification objects.

#### **4.3.158 Alias\_identification to Security\_level**

Each Alias\_identification is\_applied\_to exactly one Security\_level. Each Security\_level is related to zero, one, or more Alias\_identification objects.

#### **4.3.159 Alias\_identification to Signal**

Each Alias\_identification is\_applied\_to exactly one Signal. Each Signal is related to zero, one, or more Alias\_identification objects.

#### **4.3.160 Alias\_identification to Specification**

Each Alias\_identification is\_applied\_to exactly one Specification. Each Specification is related to zero, one, or more Alias\_identification objects.

#### **4.3.161 Alias\_identification to Specification\_category**

Each Alias\_identification is\_applied\_to exactly one Specification\_category. Each Specification\_category is related to zero, one, or more Alias\_identification objects.

#### **4.3.162 Alias\_identification to Technical\_system**

Each Alias\_identification is\_applied\_to exactly one Technical\_system. Each Technical\_system is related to zero, one, or more Alias\_identification objects.

#### **4.3.163 Alias\_identification to Terminal**

Each Alias\_identification is\_applied\_to exactly one Terminal. Each Terminal is related to zero, one, or more Alias\_identification objects.

#### **4.3.164 Alias\_version to Alias\_designation**

Each Alias\_version refers to exactly one Alias\_designation in the role of associated\_alias\_id. Each Alias\_designation acts as associated\_alias\_id for zero, one, or more Alias\_version objects.

#### **4.3.165 Alias\_version to Alias\_identification**

Each Alias\_version refers to exactly one Alias\_identification in the role of associated\_alias\_id. Each Alias\_identification acts as associated\_alias\_id for zero, one, or more Alias\_version objects.

#### **4.3.166 Alternate\_item\_relationship to Item**

Each Alternate\_item\_relationship refers to exactly one Item in the role of alternate. Each Item acts as alternate for zero, one, or more Alternate\_item\_relationship objects.

**4.3.167 Alternate\_item\_relationship to Item**

Each Alternate\_item\_relationship refers to exactly one Item in the role of base. Each Item acts as base for zero, one, or more Alternate\_item\_relationship objects.

**4.3.168 Alternative\_solution to Alternative\_solution**

Each Alternative\_solution refers to exactly one Alternative\_solution in the role of base\_element. Each Alternative\_solution acts as base\_element for zero, one, or more Alternative\_solution objects.

**4.3.169 Alternative\_solution to Function\_definition**

Each Alternative\_solution refers to exactly one Function\_definition in the role of base\_element. Each Function\_definition acts as base\_element for zero, one, or more Alternative\_solution objects.

**4.3.170 Alternative\_solution to Product\_component**

Each Alternative\_solution refers to exactly one Product\_component in the role of base\_element. Each Product\_component acts as base\_element for zero, one, or more Alternative\_solution objects.

**4.3.171 Alternative\_solution to Single\_function\_unit**

Each Alternative\_solution refers to exactly one Single\_function\_unit in the role of base\_element. Each Single\_function\_unit acts as base\_element for zero, one, or more Alternative\_solution objects.

**4.3.172 Angular\_dimension to Dimension\_line**

Each Angular\_dimension refers to exactly one Dimension\_line in the role of extent. Each Dimension\_line acts as extent for zero, one, or more Angular\_dimension objects.

**4.3.173 Angular\_dimension to Projection\_line**

Each Angular\_dimension refers to zero, one, or two Projection\_line objects in the role of component. Each Projection\_line acts as component for zero, one, or more Angular\_dimension objects.

**4.3.174 Annotation\_curve to Curve\_appearance**

Each Annotation\_curve refers to exactly one Curve\_appearance in the role of assigned\_appearance. Each Curve\_appearance acts as assigned\_appearance for zero, one, or more Annotation\_curve objects.

**4.3.175 Annotation\_subfigure to Annotation\_subfigure\_definition**

Each Annotation\_subfigure refers to exactly one Annotation\_subfigure\_definition in the role of definition. Each Annotation\_subfigure\_definition acts as definition for zero, one, or more Annotation\_subfigure objects.

#### **4.3.176 Annotation\_subfigure to Point\_2d**

Each Annotation\_subfigure refers to exactly one Point\_2d in the role of position. Each Point\_2d acts as position for zero, one, or more Annotation\_subfigure objects.

#### **4.3.177 Annotation\_subfigure\_definition to Cartesian\_coordinate\_space\_2d**

Each Annotation\_subfigure\_definition refers to exactly one Cartesian\_coordinate\_space\_2d in the role of coordinate\_space. Each Cartesian\_coordinate\_space\_2d acts as coordinate\_space for zero, one, or more Annotation\_subfigure\_definition objects.

#### **4.3.178 Annotation\_subfigure\_definition to Rectangular\_area**

Each Annotation\_subfigure\_definition refers to zero or one Rectangular\_area in the role of blanking\_box. Each Rectangular\_area acts as blanking\_box for zero, one, or more Annotation\_subfigure\_definition objects.

#### **4.3.179 Annotation\_subfigure\_definition\_element to Annotation\_placed\_annotation**

Each Annotation\_subfigure\_definition\_element refers to exactly one Annotation\_placed\_annotation in the role of used\_annotation. Each Annotation\_placed\_annotation acts as used\_annotation for zero, one, or more Annotation\_subfigure\_definition\_element objects.

#### **4.3.180 Annotation\_subfigure\_definition\_element to Annotation\_subfigure\_definition**

Each Annotation\_subfigure\_definition\_element refers to exactly one Annotation\_subfigure\_definition in the role of containing\_definition. Each Annotation\_subfigure\_definition acts as containing\_definition for zero, one, or more Annotation\_subfigure\_definition\_element objects.

#### **4.3.181 Annotation\_subfigure\_definition\_element to Layer**

Each Annotation\_subfigure\_definition\_element refers to one or more Layer objects in the role of annotation\_layers. Each Layer acts as annotation\_layers for zero, one, or more Annotation\_subfigure\_definition\_element objects.

#### **4.3.182 Annotation\_subfigure\_definition\_element to Visibility**

Each Annotation\_subfigure\_definition\_element refers to exactly one Visibility in the role of annotation\_visibility. Each Visibility acts as annotation\_visibility for zero, one, or more Annotation\_subfigure\_definition\_element objects.

#### **4.3.183 Annotation\_symbol to Colour**

Each Annotation\_symbol refers to zero or one Colour in the role of overriding\_colour. Each Colour acts as overriding\_colour for zero, one, or more Annotation\_symbol objects.



**4.3.184 Annotation\_symbol to Point\_2d**

Each Annotation\_symbol refers to exactly one Point\_2d in the role of position. Each Point\_2d acts as position for zero, one, or more Annotation\_symbol objects.

**4.3.185 Annotation\_symbol to Rectangular\_area**

Each Annotation\_symbol refers to zero or one Rectangular\_area in the role of blanking\_box. Each Rectangular\_area acts as blanking\_box for zero, one, or more Annotation\_symbol objects.

**4.3.186 Approval to Activity**

Each Approval is\_applied\_to one or more Activity objects. Each Activity is related to zero, one, or more Approval objects.

**4.3.187 Approval to Activity\_element**

Each Approval is\_applied\_to one or more Activity\_element objects. Each Activity\_element is related to zero, one, or more Approval objects.

**4.3.188 Approval to Activity\_method\_assignment**

Each Approval is\_applied\_to one or more Activity\_method\_assignment objects. Each Activity\_method\_assignment is related to zero, one, or more Approval objects.

**4.3.189 Approval to Activity\_relationship**

Each Approval is\_applied\_to one or more Activity\_relationship objects. Each Activity\_relationship is related to zero, one, or more Approval objects.

**4.3.190 Approval to Alternate\_item\_relationship**

Each Approval is\_applied\_to one or more Alternate\_item\_relationship objects. Each Alternate\_item\_relationship is related to zero, one, or more Approval objects.

**4.3.191 Approval to Alternate\_item\_relationship**

Each Approval is\_applied\_to one or more Alternate\_item\_relationship objects. Each Alternate\_item\_relationship is related to zero, one, or more Approval objects.

**4.3.192 Approval to Approval\_status**

Each Approval refers to exactly one Approval\_status in the role of status. Each Approval\_status acts as status for zero, one, or more Approval objects.

**4.3.193 Approval to Assembly\_component\_relationship**

Each Approval is\_applied\_to one or more Assembly\_component\_relationship objects. Each Assembly\_component\_relationship is related to zero, one, or more Approval objects.

#### **4.3.194 Approval to Cable\_pull\_information**

Each Approval is\_applied\_to one or more Cable\_pull\_information objects. Each Cable\_pull\_information is related to zero, one, or more Approval objects.

#### **4.3.195 Approval to Certification**

Each Approval is\_applied\_to one or more Certification objects. Each Certification is related to zero, one, or more Approval objects.

#### **4.3.196 Approval to Class\_category\_association**

Each Approval is\_applied\_to one or more Class\_category\_association objects. Each Class\_category\_association is related to zero, one, or more Approval objects.

#### **4.3.197 Approval to Class\_condition\_association**

Each Approval is\_applied\_to one or more Class\_condition\_association objects. Each Class\_condition\_association is related to zero, one, or more Approval objects.

#### **4.3.198 Approval to Class\_inclusion\_association**

Each Approval is\_applied\_to one or more Class\_inclusion\_association objects. Each Class\_inclusion\_association is related to zero, one, or more Approval objects.

#### **4.3.199 Approval to Class\_specification\_association**

Each Approval is\_applied\_to one or more Class\_specification\_association objects. Each Class\_specification\_association is related to zero, one, or more Approval objects.

#### **4.3.200 Approval to Class\_structure\_relationship**

Each Approval is\_applied\_to one or more Class\_structure\_relationship objects. Each Class\_structure\_relationship is related to zero, one, or more Approval objects.

#### **4.3.201 Approval to Classification\_association**

Each Approval is\_applied\_to one or more Classification\_association objects. Each Classification\_association is related to zero, one, or more Approval objects.

#### **4.3.202 Approval to Classification\_system**

Each Approval is\_applied\_to one or more Classification\_system objects. Each Classification\_system is related to zero, one, or more Approval objects.

#### **4.3.203 Approval to Complex\_product**

Each Approval is\_applied\_to one or more Complex\_product objects. Each Complex\_product is related to zero, one, or more Approval objects.

#### **4.3.204 Approval to Complex\_product\_relationship**

Each Approval is\_applied\_to one or more Complex\_product\_relationship objects. Each Complex\_product\_relationship is related to zero, one, or more Approval objects.

#### **4.3.205 Approval to Composition\_relationship**

Each Approval is\_applied\_to one or more Composition\_relationship objects. Each Composition\_relationship is related to zero, one, or more Approval objects.

#### **4.3.206 Approval to Configuration**

Each Approval is\_applied\_to one or more Configuration objects. Each Configuration is related to zero, one, or more Approval objects.

#### **4.3.207 Approval to Connectivity\_allocation**

Each Approval is\_applied\_to one or more Connectivity\_allocation objects. Each Connectivity\_allocation is related to zero, one, or more Approval objects.

#### **4.3.208 Approval to Connectivity\_definition**

Each Approval is\_applied\_to one or more Connectivity\_definition objects. Each Connectivity\_definition is related to zero, one, or more Approval objects.

#### **4.3.209 Approval to Connectivity\_definition\_relationship**

Each Approval is\_applied\_to one or more Connectivity\_definition\_relationship objects. Each Connectivity\_definition\_relationship is related to zero, one, or more Approval objects.

#### **4.3.210 Approval to Contract**

Each Approval is\_applied\_to one or more Contract objects. Each Contract is related to zero, one, or more Approval objects.

#### **4.3.211 Approval to Data\_element**

Each Approval is\_applied\_to one or more Data\_element objects. Each Data\_element is related to zero, one, or more Approval objects.

#### **4.3.212 Approval to Data\_element\_association**

Each Approval is\_applied\_to one or more Data\_element\_association objects. Each Data\_element\_association is related to zero, one, or more Approval objects.

#### **4.3.213 Approval to Data\_element\_definition**

Each Approval is\_applied\_to one or more Data\_element\_definition objects. Each Data\_element\_definition is related to zero, one, or more Approval objects.

#### **4.3.214 Approval to Data\_element\_relationship**

Each Approval is\_applied\_to one or more Data\_element\_relationship objects. Each Data\_element\_relationship is related to zero, one, or more Approval objects.

#### **4.3.215 Approval to Date\_and\_person\_or\_organization**

Each Approval is\_approved\_by zero, one, or more Date\_and\_person\_or\_organization objects. Each Date\_and\_person\_or\_organization is related to zero, one, or more Approval objects.

#### **4.3.216 Approval to Date\_time**

Each Approval refers to zero or one Date\_time in the role of actual\_date. Each Date\_time acts as actual\_date for zero, one, or more Approval objects.

#### **4.3.217 Approval to Date\_time**

Each Approval refers to zero or one Date\_time in the role of planned\_date. Each Date\_time acts as planned\_date for zero, one, or more Approval objects.

#### **4.3.218 Approval to Design\_discipline\_item\_definition**

Each Approval is\_applied\_to one or more Design\_discipline\_item\_definition objects. Each Design\_discipline\_item\_definition is related to zero, one, or more Approval objects.

#### **4.3.219 Approval to Device**

Each Approval is\_applied\_to one or more Device objects. Each Device is related to zero, one, or more Approval objects.

#### **4.3.220 Approval to Device\_relationship**

Each Approval is\_applied\_to one or more Device\_relationship objects. Each Device\_relationship is related to zero, one, or more Approval objects.

#### **4.3.221 Approval to Document**

Each Approval is\_applied\_to one or more Document objects. Each Document is related to zero, one, or more Approval objects.

#### **4.3.222 Approval to Document\_file**

Each Approval is\_applied\_to one or more Document\_file objects. Each Document\_file is related to zero, one, or more Approval objects.

#### **4.3.223 Approval to Document\_file\_relationship**

Each Approval is\_applied\_to one or more Document\_file\_relationship objects. Each Document\_file\_relationship is related to zero, one, or more Approval objects.

#### **4.3.224 Approval to Document\_representation**

Each Approval is\_applied\_to one or more Document\_representation objects. Each Document\_representation is related to zero, one, or more Approval objects.

#### **4.3.225 Approval to Document\_version**

Each Approval is\_applied\_to one or more Document\_version objects. Each Document\_version is related to zero, one, or more Approval objects.

#### **4.3.226 Approval to Document\_version\_relationship**

Each Approval is\_applied\_to one or more Document\_version\_relationship objects. Each Document\_version\_relationship is related to zero, one, or more Approval objects.

#### **4.3.227 Approval to Drawing**

Each Approval is\_applied\_to one or more Drawing objects. Each Drawing is related to zero, one, or more Approval objects.

#### **4.3.228 Approval to Drawing\_sequence**

Each Approval is\_applied\_to one or more Drawing\_sequence objects. Each Drawing\_sequence is related to zero, one, or more Approval objects.

#### **4.3.229 Approval to Drawing\_sheet**

Each Approval is\_applied\_to one or more Drawing\_sheet objects. Each Drawing\_sheet is related to zero, one, or more Approval objects.

#### **4.3.230 Approval to Drawing\_sheet\_relationship**

Each Approval is\_applied\_to one or more Drawing\_sheet\_relationship objects. Each Drawing\_sheet\_relationship is related to zero, one, or more Approval objects.

#### **4.3.231 Approval to Function\_definition**

Each Approval is\_applied\_to one or more Function\_definition objects. Each Function\_definition is related to zero, one, or more Approval objects.

#### **4.3.232 Approval to Function\_definition\_relationship**

Each Approval is\_applied\_to one or more Function\_definition\_relationship objects. Each Function\_definition\_relationship is related to zero, one, or more Approval objects.

#### **4.3.233 Approval to Function\_interface**

Each Approval is\_applied\_to one or more Function\_interface objects. Each Function\_interface is related to zero, one, or more Approval objects.

#### **4.3.234 Approval to Function\_unit**

Each Approval is\_applied\_to one or more Function\_unit objects. Each Function\_unit is related to zero, one, or more Approval objects.

#### **4.3.235 Approval to Function\_unit\_relationship**

Each Approval is\_applied\_to one or more Function\_unit\_relationship objects. Each Function\_unit\_relationship is related to zero, one, or more Approval objects.

#### **4.3.236 Approval to Function\_version**

Each Approval is\_applied\_to one or more Function\_version objects. Each Function\_version is related to zero, one, or more Approval objects.

#### **4.3.237 Approval to Function\_version\_relationship**

Each Approval is\_applied\_to one or more Function\_version\_relationship objects. Each Function\_version\_relationship is related to zero, one, or more Approval objects.

#### **4.3.238 Approval to Functional\_connectivity\_definition**

Each Approval is\_applied\_to one or more Functional\_connectivity\_definition objects. Each Functional\_connectivity\_definition is related to zero, one, or more Approval objects.

#### **4.3.239 Approval to Functional\_connectivity\_definition\_relationship**

Each Approval is\_applied\_to one or more Functional\_connectivity\_definition\_relationship objects. Each Functional\_connectivity\_definition\_relationship is related to zero, one, or more Approval objects.

#### **4.3.240 Approval to Functional\_unit\_allocation**

Each Approval is\_applied\_to one or more Functional\_unit\_allocation objects. Each Functional\_unit\_allocation is related to zero, one, or more Approval objects.

#### **4.3.241 Approval to General\_classification**

Each Approval is\_applied\_to one or more General\_classification objects. Each General\_classification is related to zero, one, or more Approval objects.

#### **4.3.242 Approval to Generic\_note**

Each Approval is\_applied\_to one or more Generic\_note objects. Each Generic\_note is related to zero, one, or more Approval objects.

#### **4.3.243 Approval to Interface**

Each Approval is\_applied\_to one or more Interface objects. Each Interface is related to zero, one, or more Approval objects.

#### **4.3.244 Approval to Interface\_port**

Each Approval is\_applied\_to one or more Interface\_port objects. Each Interface\_port is related to zero, one, or more Approval objects.

#### **4.3.245 Approval to Interface\_terminal**

Each Approval is\_applied\_to one or more Interface\_terminal objects. Each Interface\_terminal is related to zero, one, or more Approval objects.

#### **4.3.246 Approval to Item\_definition\_relationship**

Each Approval is\_applied\_to one or more Item\_definition\_relationship objects. Each Item\_definition\_relationship is related to zero, one, or more Approval objects.

#### **4.3.247 Approval to Item\_version**

Each Approval is\_applied\_to one or more Item\_version objects. Each Item\_version is related to zero, one, or more Approval objects.

#### **4.3.248 Approval to Item\_version\_relationship**

Each Approval is\_applied\_to one or more Item\_version\_relationship objects. Each Item\_version\_relationship is related to zero, one, or more Approval objects.

#### **4.3.249 Approval to Location**

Each Approval is\_applied\_to one or more Location objects. Each Location is related to zero, one, or more Approval objects.

#### **4.3.250 Approval to Location\_relationship**

Each Approval is\_applied\_to one or more Location\_relationship objects. Each Location\_relationship is related to zero, one, or more Approval objects.

#### **4.3.251 Approval to Manufacturing\_configuration**

Each Approval is\_applied\_to one or more Manufacturing\_configuration objects. Each Manufacturing\_configuration is related to zero, one, or more Approval objects.

#### **4.3.252 Approval to Marking**

Each Approval is\_applied\_to one or more Marking objects. Each Marking is related to zero, one, or more Approval objects.

#### **4.3.253 Approval to Material**

Each Approval is\_applied\_to one or more Material objects. Each Material is related to zero, one, or more Approval objects.

#### **4.3.254 Approval to Node**

Each Approval is\_applied\_to one or more Node objects. Each Node is related to zero, one, or more Approval objects.

#### **4.3.255 Approval to Node\_relationship**

Each Approval is\_applied\_to one or more Node\_relationship objects. Each Node\_relationship is related to zero, one, or more Approval objects.

#### **4.3.256 Approval to Notification**

Each Approval is\_applied\_to one or more Notification objects. Each Notification is related to zero, one, or more Approval objects.

#### **4.3.257 Approval to Notification\_relationship**

Each Approval is\_applied\_to one or more Notification\_relationship objects. Each Notification\_relationship is related to zero, one, or more Approval objects.

#### **4.3.258 Approval to Offered\_function\_allocation**

Each Approval is\_applied\_to one or more Offered\_function\_allocation objects. Each Offered\_function\_allocation is related to zero, one, or more Approval objects.

#### **4.3.259 Approval to Organization**

Each Approval refers to zero, one, or more Organization objects in the role of scope. Each Organization acts as scope for zero, one, or more Approval objects.

#### **4.3.260 Approval to Path**

Each Approval is\_applied\_to one or more Path objects. Each Path is related to zero, one, or more Approval objects.

#### **4.3.261 Approval to Path\_node**

Each Approval is\_applied\_to one or more Path\_node objects. Each Path\_node is related to zero, one, or more Approval objects.

#### **4.3.262 Approval to Path\_node\_relationship**

Each Approval is\_applied\_to one or more Path\_node\_relationship objects. Each Path\_node\_relationship is related to zero, one, or more Approval objects.

#### **4.3.263 Approval to Path\_relationship**

Each Approval is\_applied\_to one or more Path\_relationship objects. Each Path\_relationship is related to zero, one, or more Approval objects.



**4.3.264 Approval to Physical\_assembly\_relationship**

Each Approval is\_applied\_to one or more Physical\_assembly\_relationship objects. Each Physical\_assembly\_relationship is related to zero, one, or more Approval objects.

**4.3.265 Approval to Physical\_instance**

Each Approval is\_applied\_to one or more Physical\_instance objects. Each Physical\_instance is related to zero, one, or more Approval objects.

**4.3.266 Approval to Port**

Each Approval is\_applied\_to one or more Port objects. Each Port is related to zero, one, or more Approval objects.

**4.3.267 Approval to Port\_allocation**

Each Approval is\_applied\_to one or more Port\_allocation objects. Each Port\_allocation is related to zero, one, or more Approval objects.

**4.3.268 Approval to Preferred\_item\_allocation**

Each Approval is\_applied\_to one or more Preferred\_item\_allocation objects. Each Preferred\_item\_allocation is related to zero, one, or more Approval objects.

**4.3.269 Approval to Preferred\_item\_terminal\_allocation**

Each Approval is\_applied\_to one or more Preferred\_item\_terminal\_allocation objects. Each Preferred\_item\_terminal\_allocation is related to zero, one, or more Approval objects.

**4.3.270 Approval to Process\_variable**

Each Approval is\_applied\_to one or more Process\_variable objects. Each Process\_variable is related to zero, one, or more Approval objects.

**4.3.271 Approval to Process\_variable\_relationship**

Each Approval is\_applied\_to one or more Process\_variable\_relationship objects. Each Process\_variable\_relationship is related to zero, one, or more Approval objects.

**4.3.272 Approval to Product\_class**

Each Approval is\_applied\_to one or more Product\_class objects. Each Product\_class is related to zero, one, or more Approval objects.

**4.3.273 Approval to Product\_identification**

Each Approval is\_applied\_to one or more Product\_identification objects. Each Product\_identification is related to zero, one, or more Approval objects.

#### **4.3.274 Approval to Product\_structure\_relationship**

Each Approval is\_applied\_to one or more Product\_structure\_relationship objects. Each Product\_structure\_relationship is related to zero, one, or more Approval objects.

#### **4.3.275 Approval to Project**

Each Approval is\_applied\_to one or more Project objects. Each Project is related to zero, one, or more Approval objects.

#### **4.3.276 Approval to Requirement**

Each Approval is\_applied\_to one or more Requirement objects. Each Requirement is related to zero, one, or more Approval objects.

#### **4.3.277 Approval to Route**

Each Approval is\_applied\_to one or more Route objects. Each Route is related to zero, one, or more Approval objects.

#### **4.3.278 Approval to Route\_relationship**

Each Approval is\_applied\_to one or more Route\_relationship objects. Each Route\_relationship is related to zero, one, or more Approval objects.

#### **4.3.279 Approval to Section**

Each Approval is\_applied\_to one or more Section objects. Each Section is related to zero, one, or more Approval objects.

#### **4.3.280 Approval to Section\_end**

Each Approval is\_applied\_to one or more Section\_end objects. Each Section\_end is related to zero, one, or more Approval objects.

#### **4.3.281 Approval to Section\_interface**

Each Approval is\_applied\_to one or more Section\_interface objects. Each Section\_interface is related to zero, one, or more Approval objects.

#### **4.3.282 Approval to Section\_interface\_relationship**

Each Approval is\_applied\_to one or more Section\_interface\_relationship objects. Each Section\_interface\_relationship is related to zero, one, or more Approval objects.

#### **4.3.283 Approval to Section\_relationship**

Each Approval is\_applied\_to one or more Section\_relationship objects. Each Section\_relationship is related to zero, one, or more Approval objects.

#### **4.3.284 Approval to Security\_classification**

Each Approval is\_applied\_to one or more Security\_classification objects. Each Security\_classification is related to zero, one, or more Approval objects.

#### **4.3.285 Approval to Signal**

Each Approval is\_applied\_to one or more Signal objects. Each Signal is related to zero, one, or more Approval objects.

#### **4.3.286 Approval to Signal\_relationship**

Each Approval is\_applied\_to one or more Signal\_relationship objects. Each Signal\_relationship is related to zero, one, or more Approval objects.

#### **4.3.287 Approval to Signal\_value**

Each Approval is\_applied\_to one or more Signal\_value objects. Each Signal\_value is related to zero, one, or more Approval objects.

#### **4.3.288 Approval to Specification**

Each Approval is\_applied\_to one or more Specification objects. Each Specification is related to zero, one, or more Approval objects.

#### **4.3.289 Approval to Specification\_category**

Each Approval is\_applied\_to one or more Specification\_category objects. Each Specification\_category is related to zero, one, or more Approval objects.

#### **4.3.290 Approval to Specification\_expression**

Each Approval is\_applied\_to one or more Specification\_expression objects. Each Specification\_expression is related to zero, one, or more Approval objects.

#### **4.3.291 Approval to Specification\_inclusion**

Each Approval is\_applied\_to one or more Specification\_inclusion objects. Each Specification\_inclusion is related to zero, one, or more Approval objects.

#### **4.3.292 Approval to Technical\_system**

Each Approval is\_applied\_to one or more Technical\_system objects. Each Technical\_system is related to zero, one, or more Approval objects.

#### **4.3.293 Approval to Technical\_system\_relationship**

Each Approval is\_applied\_to one or more Technical\_system\_relationship objects. Each Technical\_system\_relationship is related to zero, one, or more Approval objects.

#### **4.3.294 Approval to Terminal**

Each Approval is\_applied\_to one or more Terminal objects. Each Terminal is related to zero, one, or more Approval objects.

#### **4.3.295 Approval to Work\_order**

Each Approval is\_applied\_to one or more Work\_order objects. Each Work\_order is related to zero, one, or more Approval objects.

#### **4.3.296 Approval to Work\_request**

Each Approval is\_applied\_to one or more Work\_request objects. Each Work\_request is related to zero, one, or more Approval objects.

#### **4.3.297 Approval\_relationship to Approval**

Each Approval\_relationship refers to exactly one Approval in the role of related. Each Approval acts as related for zero, one, or more Approval\_relationship objects.

#### **4.3.298 Approval\_relationship to Approval**

Each Approval\_relationship refers to exactly one Approval in the role of relating. Each Approval acts as relating for zero, one, or more Approval\_relationship objects.

#### **4.3.299 Approval\_status to Classification\_system**

Each Approval\_status refers to zero or one Classification\_system in the role of used\_classification\_system. Each Classification\_system acts as used\_classification\_system for zero, one, or more Approval\_status objects.

#### **4.3.300 Assembly\_component\_relationship to Assembly\_definition**

Each Assembly\_component\_relationship refers to exactly one Assembly\_definition in the role of relating. Each Assembly\_definition acts as relating for zero, one, or more Assembly\_component\_relationship objects.

#### **4.3.301 Assembly\_component\_relationship to Device**

Each Assembly\_component\_relationship refers to exactly one Device in the role of related. Each Device acts as related for zero, one, or more Assembly\_component\_relationship objects.

#### **4.3.302 Assembly\_substitute\_relationship to Assembly\_component\_relationship**

Each Assembly\_substitute\_relationship refers to exactly one Assembly\_component\_relationship in the role of base. Each Assembly\_component\_relationship acts as base for zero, one, or more Assembly\_substitute\_relationship objects.

### **4.3.303 Assembly\_substitute\_relationship to Assembly\_component\_-relationship**

Each Assembly\_substitute\_relationship refers to exactly one Assembly\_component\_relationship in the role of substitute. Each Assembly\_component\_relationship acts as substitute for zero, one, or more Assembly\_substitute\_relationship objects.

### **4.3.304 Body\_breadth to Value\_with\_unit**

Each Body\_breadth refers to exactly one Value\_with\_unit in the role of value\_of\_body\_breadth. Each Value\_with\_unit acts as value\_of\_body\_breadth for zero, one, or more Body\_breadth objects.

### **4.3.305 Body\_height to Value\_with\_unit**

Each Body\_height refers to exactly one Value\_with\_unit in the role of value\_of\_body\_height. Each Value\_with\_unit acts as value\_of\_body\_height for zero, one, or more Body\_height objects.

### **4.3.306 Body\_length to Value\_with\_unit**

Each Body\_length refers to exactly one Value\_with\_unit in the role of value\_of\_body\_length. Each Value\_with\_unit acts as value\_of\_body\_length for zero, one, or more Body\_length objects.

### **4.3.307 Cable\_pull\_information to Routed\_object**

Each Cable\_pull\_information refers to one or more Routed\_object objects in the role of associated\_object. Each Routed\_object acts as associated\_object for zero, one, or more Cable\_pull\_information objects.

### **4.3.308 Cartesian\_coordinate\_space\_2d to Numerical\_precision**

Each Cartesian\_coordinate\_space\_2d refers to exactly one Numerical\_precision in the role of precision. Each Numerical\_precision acts as precision for zero, one, or more Cartesian\_coordinate\_space\_2d objects.

### **4.3.309 Cartesian\_coordinate\_space\_3d to Numerical\_precision**

Each Cartesian\_coordinate\_space\_3d refers to exactly one Numerical\_precision in the role of precision. Each Numerical\_precision acts as precision for zero, one, or more Cartesian\_coordinate\_space\_3d objects.

### **4.3.310 Certification to Device**

Each Certification is\_applied\_to one or more Device objects. Each Device is related to zero, one, or more Certification objects.

### **4.3.311 Certification to Item\_version**

Each Certification is\_applied\_to one or more Item\_version objects. Each Item\_version is related to zero, one, or more Certification objects.

#### **4.3.312 Certification to Item\_version\_relationship**

Each Certification is\_applied\_to one or more Item\_version\_relationship objects. Each Item\_version\_relationship is related to zero, one, or more Certification objects.

#### **4.3.313 Certification to Supplier\_solution**

Each Certification is\_applied\_to one or more Supplier\_solution objects. Each Supplier\_solution is related to zero, one, or more Certification objects.

#### **4.3.314 Class\_category\_association to Product\_class**

Each Class\_category\_association refers to exactly one Product\_class in the role of associated\_product\_class. Each Product\_class acts as associated\_product\_class for zero, one, or more Class\_category\_association objects.

#### **4.3.315 Class\_category\_association to Specification\_category**

Each Class\_category\_association refers to exactly one Specification\_category in the role of associated\_category. Each Specification\_category acts as associated\_category for zero, one, or more Class\_category\_association objects.

#### **4.3.316 Class\_condition\_association to Product\_class**

Each Class\_condition\_association refers to exactly one Product\_class in the role of associated\_product\_class. Each Product\_class acts as associated\_product\_class for zero, one, or more Class\_condition\_association objects.

#### **4.3.317 Class\_condition\_association to Specification\_expression**

Each Class\_condition\_association refers to exactly one Specification\_expression in the role of associated\_condition. Each Specification\_expression acts as associated\_condition for zero, one, or more Class\_condition\_association objects.

#### **4.3.318 Class\_inclusion\_association to Product\_class**

Each Class\_inclusion\_association refers to exactly one Product\_class in the role of associated\_product\_class. Each Product\_class acts as associated\_product\_class for zero, one, or more Class\_inclusion\_association objects.

#### **4.3.319 Class\_inclusion\_association to Specification\_inclusion**

Each Class\_inclusion\_association refers to exactly one Specification\_inclusion in the role of associated\_inclusion. Each Specification\_inclusion acts as associated\_inclusion for zero, one, or more Class\_inclusion\_association objects.

**4.3.320 Class\_specification\_association to Product\_class**

Each Class\_specification\_association refers to exactly one Product\_class in the role of associated\_product\_class. Each Product\_class acts as associated\_product\_class for zero, one, or more Class\_specification\_association objects.

**4.3.321 Class\_specification\_association to Specification**

Each Class\_specification\_association refers to exactly one Specification in the role of associated\_specification. Each Specification acts as associated\_specification for zero, one, or more Class\_specification\_association objects.

**4.3.322 Class\_structure\_relationship to Function\_definition**

Each Class\_structure\_relationship refers to exactly one Function\_definition in the role of related. Each Function\_definition acts as related for zero, one, or more Class\_structure\_relationship objects.

**4.3.323 Class\_structure\_relationship to Product\_class**

Each Class\_structure\_relationship refers to exactly one Product\_class in the role of relating. Each Product\_class acts as relating for zero, one, or more Class\_structure\_relationship objects.

**4.3.324 Class\_structure\_relationship to Product\_component**

Each Class\_structure\_relationship refers to exactly one Product\_component in the role of related. Each Product\_component acts as related for zero, one, or more Class\_structure\_relationship objects.

**4.3.325 Classification\_association to Activity**

Each Classification\_association refers to exactly one Activity in the role of classified\_element. Each Activity acts as classified\_element for zero, one, or more Classification\_association objects.

**4.3.326 Classification\_association to Activity\_method**

Each Classification\_association refers to exactly one Activity\_method in the role of classified\_element. Each Activity\_method acts as classified\_element for zero, one, or more Classification\_association objects.

**4.3.327 Classification\_association to Annotation\_subfigure\_definition**

Each Classification\_association refers to exactly one Annotation\_subfigure\_definition in the role of classified\_element. Each Annotation\_subfigure\_definition acts as classified\_element for zero, one, or more Classification\_association objects.

**4.3.328 Classification\_association to Approval**

Each Classification\_association refers to exactly one Approval in the role of classified\_element. Each Approval acts as classified\_element for zero, one, or more Classification\_association objects.

**4.3.329 Classification\_association to Approval\_status**

## **ISO 10303-212:2001(E)**

Each Classification\_association refers to exactly one Approval\_status in the role of classified\_element. Each Approval\_status acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.330 Classification\_association to Complex\_product**

Each Classification\_association refers to exactly one Complex\_product in the role of classified\_element. Each Complex\_product acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.331 Classification\_association to Connectivity\_definition**

Each Classification\_association refers to exactly one Connectivity\_definition in the role of classified\_element. Each Connectivity\_definition acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.332 Classification\_association to Contract**

Each Classification\_association refers to exactly one Contract in the role of classified\_element. Each Contract acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.333 Classification\_association to Data\_element**

Each Classification\_association refers to exactly one Data\_element in the role of classified\_element. Each Data\_element acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.334 Classification\_association to Data\_element\_association**

Each Classification\_association refers to exactly one Data\_element\_association in the role of classified\_element. Each Data\_element\_association acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.335 Classification\_association to Data\_element\_definition**

Each Classification\_association refers to exactly one Data\_element\_definition in the role of classified\_element. Each Data\_element\_definition acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.336 Classification\_association to Design\_discipline\_item\_definition**

Each Classification\_association refers to exactly one Design\_discipline\_item\_definition in the role of classified\_element. Each Design\_discipline\_item\_definition acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.337 Classification\_association to Device**

Each Classification\_association refers to exactly one Device in the role of classified\_element. Each Device acts as classified\_element for zero, one, or more Classification\_association objects.



#### **4.3.338 Classification\_association to Document**

Each Classification\_association refers to exactly one Document in the role of classified\_element. Each Document acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.339 Classification\_association to Document\_file**

Each Classification\_association refers to exactly one Document\_file in the role of classified\_element. Each Document\_file acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.340 Classification\_association to Document\_version**

Each Classification\_association refers to exactly one Document\_version in the role of classified\_element. Each Document\_version acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.341 Classification\_association to Drawing**

Each Classification\_association refers to exactly one Drawing in the role of classified\_element. Each Drawing acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.342 Classification\_association to Drawing\_sheet**

Each Classification\_association refers to exactly one Drawing\_sheet in the role of classified\_element. Each Drawing\_sheet acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.343 Classification\_association to Function\_definition**

Each Classification\_association refers to exactly one Function\_definition in the role of classified\_element. Each Function\_definition acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.344 Classification\_association to Function\_unit**

Each Classification\_association refers to exactly one Function\_unit in the role of classified\_element. Each Function\_unit acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.345 Classification\_association to Function\_version**

Each Classification\_association refers to exactly one Function\_version in the role of classified\_element. Each Function\_version acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.346 Classification\_association to Functional\_connectivity\_definition**

Each Classification\_association refers to exactly one Functional\_connectivity\_definition in the role of classified\_element. Each Functional\_connectivity\_definition acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.347 Classification\_association to Functionality**

Each Classification\_association refers to exactly one Functionality in the role of classified\_element. Each Functionality acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.348 Classification\_association to General\_classification**

Each Classification\_association refers to exactly one General\_classification in the role of classification. Each General\_classification acts as classification for zero, one, or more Classification\_association objects.

#### **4.3.349 Classification\_association to Interface\_port**

Each Classification\_association refers to exactly one Interface\_port in the role of classified\_element. Each Interface\_port acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.350 Classification\_association to Interface\_terminal**

Each Classification\_association refers to exactly one Interface\_terminal in the role of classified\_element. Each Interface\_terminal acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.351 Classification\_association to Item**

Each Classification\_association refers to exactly one Item in the role of classified\_element. Each Item acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.352 Classification\_association to Item\_version**

Each Classification\_association refers to exactly one Item\_version in the role of classified\_element. Each Item\_version acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.353 Classification\_association to Location**

Each Classification\_association refers to exactly one Location in the role of classified\_element. Each Location acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.354 Classification\_association to Path**

Each Classification\_association refers to exactly one Path in the role of classified\_element. Each Path acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.355 Classification\_association to Path\_node**

Each Classification\_association refers to exactly one Path\_node in the role of classified\_element. Each Path\_node acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.356 Classification\_association to Port**

Each Classification\_association refers to exactly one Port in the role of classified\_element. Each Port acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.357 Classification\_association to Product\_class**

Each Classification\_association refers to exactly one Product\_class in the role of classified\_element. Each Product\_class acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.358 Classification\_association to Product\_identification**

Each Classification\_association refers to exactly one Product\_identification in the role of classified\_element. Each Product\_identification acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.359 Classification\_association to Project**

Each Classification\_association refers to exactly one Project in the role of classified\_element. Each Project acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.360 Classification\_association to Requirement**

Each Classification\_association refers to exactly one Requirement in the role of classified\_element. Each Requirement acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.361 Classification\_association to Route**

Each Classification\_association refers to exactly one Route in the role of classified\_element. Each Route acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.362 Classification\_association to Section**

Each Classification\_association refers to exactly one Section in the role of classified\_element. Each Section acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.363 Classification\_association to Section\_interface**

Each Classification\_association refers to exactly one Section\_interface in the role of classified\_element. Each Section\_interface acts as classified\_element for zero, one, or more Classification\_association objects.

#### **4.3.364 Classification\_association to Security\_level**

## **ISO 10303-212:2001(E)**

Each Classification\_association refers to exactly one Security\_level in the role of classified\_element. Each Security\_level acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.365 Classification\_association to Signal**

Each Classification\_association refers to exactly one Signal in the role of classified\_element. Each Signal acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.366 Classification\_association to Specification\_category**

Each Classification\_association refers to exactly one Specification\_category in the role of classified\_element. Each Specification\_category acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.367 Classification\_association to Terminal**

Each Classification\_association refers to exactly one Terminal in the role of classified\_element. Each Terminal acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.368 Classification\_association to Typical\_schematic\_node**

Each Classification\_association refers to exactly one Typical\_schematic\_node in the role of classified\_element. Each Typical\_schematic\_node acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.369 Classification\_association to User\_defined\_symbol\_definition**

Each Classification\_association refers to exactly one User\_defined\_symbol\_definition in the role of classified\_element. Each User\_defined\_symbol\_definition acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.370 Classification\_association to Work\_order**

Each Classification\_association refers to exactly one Work\_order in the role of classified\_element. Each Work\_order acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.371 Classification\_association to Work\_request**

Each Classification\_association refers to exactly one Work\_request in the role of classified\_element. Each Work\_request acts as classified\_element for zero, one, or more Classification\_association objects.

### **4.3.372 Classification\_attribute to Data\_element**

Each Classification\_attribute refers to zero, one, or more Data\_element objects in the role of allowed\_value. Each Data\_element acts as allowed\_value for zero, one, or more Classification\_attribute objects.

### **4.3.373 Classification\_attribute to Data\_element\_definition**

Each Classification\_attribute refers to exactly one Data\_element\_definition in the role of attribute\_definition. Each Data\_element\_definition acts as attribute\_definition for zero, one, or more Classification\_attribute objects.

#### **4.3.374 Classification\_attribute to General\_classification**

Each Classification\_attribute refers to exactly one General\_classification in the role of associated\_classification. Each General\_classification acts as associated\_classification for zero, one, or more Classification\_attribute objects.

#### **4.3.375 Coded\_size to Classification\_system**

Each Coded\_size refers to exactly one Classification\_system in the role of referenced\_standard. Each Classification\_system acts as referenced\_standard for zero, one, or more Coded\_size objects.

#### **4.3.376 Complex\_product\_relationship to Complex\_product**

Each Complex\_product\_relationship refers to exactly one Complex\_product in the role of related. Each Complex\_product acts as related for zero, one, or more Complex\_product\_relationship objects.

#### **4.3.377 Complex\_product\_relationship to Complex\_product**

Each Complex\_product\_relationship refers to exactly one Complex\_product in the role of relating. Each Complex\_product acts as relating for zero, one, or more Complex\_product\_relationship objects.

#### **4.3.378 Component\_colour to Classification\_system**

Each Component\_colour refers to zero or one Classification\_system in the role of coding\_system. Each Classification\_system acts as coding\_system for zero, one, or more Component\_colour objects.

#### **4.3.379 Component\_placement to Product\_component**

Each Component\_placement refers to exactly one Product\_component in the role of placed\_component. Each Product\_component acts as placed\_component for zero, one, or more Component\_placement objects.

#### **4.3.380 Component\_placement to Product\_component**

Each Component\_placement refers to exactly one Product\_component in the role of reference\_product\_component. Each Product\_component acts as reference\_product\_component for zero, one, or more Component\_placement objects.

#### **4.3.381 Composition\_relationship to Function\_definition**

Each Composition\_relationship refers to exactly one Function\_definition in the role of composed\_function. Each Function\_definition acts as composed\_function for zero, one, or more Composition\_relationship objects.

#### **4.3.382 Composition\_relationship to Function\_unit**

Each Composition\_relationship refers to exactly one Function\_unit in the role of functional\_component. Each Function\_unit acts as functional\_component for zero, one, or more Composition\_relationship objects.

#### **4.3.383 Configuration to Class\_condition\_association**

Each Configuration is\_solution\_for exactly one Class\_condition\_association. Each Class\_condition\_association is related to zero, one, or more Configuration objects.

#### **4.3.384 Configuration to Class\_specification\_association**

Each Configuration is\_solution\_for exactly one Class\_specification\_association. Each Class\_specification\_association is related to zero, one, or more Configuration objects.

#### **4.3.385 Configuration to Complex\_product**

Each Configuration refers to exactly one Complex\_product in the role of configured\_element. Each Complex\_product acts as configured\_element for zero, one, or more Configuration objects.

#### **4.3.386 Configuration to Connectivity\_definition**

Each Configuration refers to exactly one Connectivity\_definition in the role of configured\_element. Each Connectivity\_definition acts as configured\_element for zero, one, or more Configuration objects.

#### **4.3.387 Configuration to Device**

Each Configuration refers to exactly one Device in the role of configured\_element. Each Device acts as configured\_element for zero, one, or more Configuration objects.

#### **4.3.388 Configuration to Function\_unit**

Each Configuration refers to exactly one Function\_unit in the role of configured\_element. Each Function\_unit acts as configured\_element for zero, one, or more Configuration objects.

#### **4.3.389 Configuration to Functional\_connectivity\_definition**

Each Configuration refers to exactly one Functional\_connectivity\_definition in the role of configured\_element. Each Functional\_connectivity\_definition acts as configured\_element for zero, one, or more Configuration objects.

#### **4.3.390 Configuration to Location**

Each Configuration refers to exactly one Location in the role of configured\_element. Each Location acts as configured\_element for zero, one, or more Configuration objects.

**4.3.391 Configuration to Signal**

Each Configuration refers to exactly one Signal in the role of configured\_element. Each Signal acts as configured\_element for zero, one, or more Configuration objects.

**4.3.392 Connect\_area to Curve\_2d**

Each Connect\_area refers to one or more Curve\_2d objects in the role of defined\_by. Each Curve\_2d acts as defined\_by for zero, one, or more Connect\_area objects.

**4.3.393 Connect\_area to Point\_2d**

Each Connect\_area refers to one or more Point\_2d objects in the role of defined\_by. Each Point\_2d acts as defined\_by for zero, one, or more Connect\_area objects.

**4.3.394 Connecting\_line to Connectivity\_definition**

Each Connecting\_line refers to exactly one Connectivity\_definition in the role of presents. Each Connectivity\_definition acts as presents for zero, one, or more Connecting\_line objects.

**4.3.395 Connecting\_line to Functional\_connectivity\_definition**

Each Connecting\_line refers to exactly one Functional\_connectivity\_definition in the role of presents. Each Functional\_connectivity\_definition acts as presents for zero, one, or more Connecting\_line objects.

**4.3.396 Connection to Terminal**

Each Connection refers to zero, one, or more Terminal objects in the role of connected\_terminal. Each Terminal acts as connected\_terminal for zero, one, or more Connection objects.

**4.3.397 Connectivity\_allocation to Connectivity\_definition**

Each Connectivity\_allocation refers to exactly one Connectivity\_definition in the role of connectivity\_implementation. Each Connectivity\_definition acts as connectivity\_implementation for zero, one, or more Connectivity\_allocation objects.

**4.3.398 Connectivity\_allocation to Device**

Each Connectivity\_allocation refers to exactly one Device in the role of connectivity\_implementation. Each Device acts as connectivity\_implementation for zero, one, or more Connectivity\_allocation objects.

**4.3.399 Connectivity\_allocation to Function\_unit**

Each Connectivity\_allocation refers to exactly one Function\_unit in the role of connectivity\_implementation. Each Function\_unit acts as connectivity\_implementation for zero, one, or more Connectivity\_allocation objects.

**4.3.400 Connectivity\_allocation to Functional\_connectivity\_definition**

## **ISO 10303-212:2001(E)**

Each Connectivity\_allocation refers to exactly one Functional\_connectivity\_definition in the role of allocated\_connectivity\_definition. Each Functional\_connectivity\_definition acts as allocated\_connectivity\_definition for zero, one, or more Connectivity\_allocation objects.

### **4.3.401 Connectivity\_allocation to Physical\_instance**

Each Connectivity\_allocation refers to exactly one Physical\_instance in the role of connectivity\_implementation. Each Physical\_instance acts as connectivity\_implementation for zero, one, or more Connectivity\_allocation objects.

### **4.3.402 Connectivity\_definition to Assembly\_definition**

Each Connectivity\_definition refers to exactly one Assembly\_definition in the role of connectivity\_of. Each Assembly\_definition acts as connectivity\_of for zero, one, or more Connectivity\_definition objects.

### **4.3.403 Connectivity\_definition to Device**

Each Connectivity\_definition refers to zero, one, or more Device objects in the role of implemented\_by. Each Device acts as implemented\_by for zero, one, or more Connectivity\_definition objects.

### **4.3.404 Connectivity\_definition to Physical\_instance**

Each Connectivity\_definition refers to zero, one, or more Physical\_instance objects in the role of implemented\_by. Each Physical\_instance acts as implemented\_by for zero, one, or more Connectivity\_definition objects.

### **4.3.405 Connectivity\_definition\_relationship to Connectivity\_definition**

Each Connectivity\_definition\_relationship refers to exactly one Connectivity\_definition in the role of related. Each Connectivity\_definition acts as related for zero, one, or more Connectivity\_definition\_relationship objects.

### **4.3.406 Connectivity\_definition\_relationship to Connectivity\_definition**

Each Connectivity\_definition\_relationship refers to exactly one Connectivity\_definition in the role of relating. Each Connectivity\_definition acts as relating for zero, one, or more Connectivity\_definition\_relationship objects.

### **4.3.407 Contract to Activity**

Each Contract refers to zero, one, or more Activity objects in the role of contracted\_element. Each Activity acts as contracted\_element for zero, one, or more Contract objects.

### **4.3.408 Contract to Data\_element**

Each Contract refers to zero, one, or more Data\_element objects in the role of contracted\_element. Each Data\_element acts as contracted\_element for zero, one, or more Contract objects.

### **4.3.409 Contract to Design\_discipline\_item\_definition**



Each Contract refers to zero, one, or more Design\_discipline\_item\_definition objects in the role of contracted\_element. Each Design\_discipline\_item\_definition acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.410 Contract to Device**

Each Contract refers to zero, one, or more Device objects in the role of contracted\_element. Each Device acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.411 Contract to Function\_definition**

Each Contract refers to zero, one, or more Function\_definition objects in the role of contracted\_element. Each Function\_definition acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.412 Contract to Function\_unit**

Each Contract refers to zero, one, or more Function\_unit objects in the role of contracted\_element. Each Function\_unit acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.413 Contract to Function\_version**

Each Contract refers to zero, one, or more Function\_version objects in the role of contracted\_element. Each Function\_version acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.414 Contract to Item\_version**

Each Contract refers to zero, one, or more Item\_version objects in the role of contracted\_element. Each Item\_version acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.415 Contract to Location**

Each Contract refers to zero, one, or more Location objects in the role of contracted\_element. Each Location acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.416 Contract to Node**

Each Contract refers to zero, one, or more Node objects in the role of contracted\_element. Each Node acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.417 Contract to Notification**

Each Contract refers to zero, one, or more Notification objects in the role of contracted\_element. Each Notification acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.418 Contract to Path**

Each Contract refers to zero, one, or more Path objects in the role of contracted\_element. Each Path acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.419 Contract to Path\_node**

Each Contract refers to zero, one, or more Path\_node objects in the role of contracted\_element. Each Path\_node acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.420 Contract to Physical\_instance**

Each Contract refers to zero, one, or more Physical\_instance objects in the role of contracted\_element. Each Physical\_instance acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.421 Contract to Process\_variable**

Each Contract refers to zero, one, or more Process\_variable objects in the role of contracted\_element. Each Process\_variable acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.422 Contract to Project**

Each Contract refers to zero, one, or more Project objects in the role of contracted\_element. Each Project acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.423 Contract to Route**

Each Contract refers to zero, one, or more Route objects in the role of contracted\_element. Each Route acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.424 Contract to Section**

Each Contract refers to zero, one, or more Section objects in the role of contracted\_element. Each Section acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.425 Contract to Section\_end**

Each Contract refers to zero, one, or more Section\_end objects in the role of contracted\_element. Each Section\_end acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.426 Contract to Section\_interface**

Each Contract refers to zero, one, or more Section\_interface objects in the role of contracted\_element. Each Section\_interface acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.427 Contract to Signal**

Each Contract refers to zero, one, or more Signal objects in the role of contracted\_element. Each Signal acts as contracted\_element for zero, one, or more Contract objects.

#### **4.3.428 Contract to Signal\_value**

Each Contract refers to zero, one, or more Signal\_value objects in the role of contracted\_element. Each Signal\_value acts as contracted\_element for zero, one, or more Contract objects.

**4.3.429 Contract to Technical\_system**

Each Contract refers to zero, one, or more Technical\_system objects in the role of contracted\_element. Each Technical\_system acts as contracted\_element for zero, one, or more Contract objects.

**4.3.430 Contract to Work\_order**

Each Contract refers to zero, one, or more Work\_order objects in the role of contracted\_element. Each Work\_order acts as contracted\_element for zero, one, or more Contract objects.

**4.3.431 Contract to Work\_request**

Each Contract refers to zero, one, or more Work\_request objects in the role of contracted\_element. Each Work\_request acts as contracted\_element for zero, one, or more Contract objects.

**4.3.432 Cross\_section to Value\_with\_unit**

Each Cross\_section refers to exactly one Value\_with\_unit in the role of value\_of\_cross\_section. Each Value\_with\_unit acts as value\_of\_cross\_section for zero, one, or more Cross\_section objects.

**4.3.433 Curve\_appearance to Colour**

Each Curve\_appearance refers to exactly one Colour in the role of curve\_colour. Each Colour acts as curve\_colour for zero, one, or more Curve\_appearance objects.

**4.3.434 Curve\_appearance to Line\_font**

Each Curve\_appearance refers to exactly one Line\_font in the role of font. Each Line\_font acts as font for zero, one, or more Curve\_appearance objects.

**4.3.435 Curve\_dimension to Dimension\_line**

Each Curve\_dimension refers to exactly one Dimension\_line in the role of extent. Each Dimension\_line acts as extent for zero, one, or more Curve\_dimension objects.

**4.3.436 Curve\_dimension to Projection\_line**

Each Curve\_dimension refers to zero, one, or two Projection\_line objects in the role of component. Each Projection\_line acts as component for zero, one, or more Curve\_dimension objects.

**4.3.437 Data\_element\_association to Activity**

Each Data\_element\_association refers to exactly one Activity in the role of associated\_item. Each Activity acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.438 Data\_element\_association to Assembly\_component\_relationship**

Each Data\_element\_association refers to exactly one Assembly\_component\_relationship in the role of associated\_item. Each Assembly\_component\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.439 Data\_element\_association to Cable\_pull\_information**

Each Data\_element\_association refers to exactly one Cable\_pull\_information in the role of associated\_item. Each Cable\_pull\_information acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.440 Data\_element\_association to Complex\_product**

Each Data\_element\_association refers to exactly one Complex\_product in the role of associated\_item. Each Complex\_product acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.441 Data\_element\_association to Complex\_product\_relationship**

Each Data\_element\_association refers to exactly one Complex\_product\_relationship in the role of associated\_item. Each Complex\_product\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.442 Data\_element\_association to Composition\_relationship**

Each Data\_element\_association refers to exactly one Composition\_relationship in the role of associated\_item. Each Composition\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.443 Data\_element\_association to Connectivity\_allocation**

Each Data\_element\_association refers to exactly one Connectivity\_allocation in the role of associated\_item. Each Connectivity\_allocation acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.444 Data\_element\_association to Connectivity\_definition**

Each Data\_element\_association refers to exactly one Connectivity\_definition in the role of associated\_item. Each Connectivity\_definition acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.445 Data\_element\_association to Connectivity\_definition\_relationship**

Each Data\_element\_association refers to exactly one Connectivity\_definition\_relationship in the role of associated\_item. Each Connectivity\_definition\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.446 Data\_element\_association to Data\_element**

Each Data\_element\_association refers to exactly one Data\_element in the role of associated\_data\_element. Each Data\_element acts as associated\_data\_element for zero, one, or more Data\_element\_association objects.

**4.3.447 Data\_element\_association to Design\_discipline\_item\_definition**

Each Data\_element\_association refers to exactly one Design\_discipline\_item\_definition in the role of associated\_item. Each Design\_discipline\_item\_definition acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.448 Data\_element\_association to Device**

Each Data\_element\_association refers to exactly one Device in the role of associated\_item. Each Device acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.449 Data\_element\_association to Device\_relationship**

Each Data\_element\_association refers to exactly one Device\_relationship in the role of associated\_item. Each Device\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.450 Data\_element\_association to Document**

Each Data\_element\_association refers to exactly one Document in the role of associated\_item. Each Document acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.451 Data\_element\_association to Document\_file**

Each Data\_element\_association refers to exactly one Document\_file in the role of associated\_item. Each Document\_file acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.452 Data\_element\_association to Document\_representation**

Each Data\_element\_association refers to exactly one Document\_representation in the role of associated\_item. Each Document\_representation acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.453 Data\_element\_association to Document\_version**

Each Data\_element\_association refers to exactly one Document\_version in the role of associated\_item. Each Document\_version acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.454 Data\_element\_association to Drawing\_sheet**

Each Data\_element\_association refers to exactly one Drawing\_sheet in the role of associated\_item. Each Drawing\_sheet acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.455 Data\_element\_association to Free\_segment**

Each Data\_element\_association refers to exactly one Free\_segment in the role of associated\_item. Each Free\_segment acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.456 Data\_element\_association to Function\_definition**

Each Data\_element\_association refers to exactly one Function\_definition in the role of associated\_item. Each Function\_definition acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.457 Data\_element\_association to Function\_definition\_relationship**

Each Data\_element\_association refers to exactly one Function\_definition\_relationship in the role of associated\_item. Each Function\_definition\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.458 Data\_element\_association to Function\_interface**

Each Data\_element\_association refers to exactly one Function\_interface in the role of associated\_item. Each Function\_interface acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.459 Data\_element\_association to Function\_unit**

Each Data\_element\_association refers to exactly one Function\_unit in the role of associated\_item. Each Function\_unit acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.460 Data\_element\_association to Function\_unit\_relationship**

Each Data\_element\_association refers to exactly one Function\_unit\_relationship in the role of associated\_item. Each Function\_unit\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.461 Data\_element\_association to Function\_version**

Each Data\_element\_association refers to exactly one Function\_version in the role of associated\_item. Each Function\_version acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.462 Data\_element\_association to Functional\_connectivity\_definition**

Each Data\_element\_association refers to exactly one Functional\_connectivity\_definition in the role of associated\_item. Each Functional\_connectivity\_definition acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.463 Data\_element\_association to Functional\_connectivity\_definition\_relationship**

Each Data\_element\_association refers to exactly one Functional\_connectivity\_definition\_relationship in the role of associated\_item. Each Functional\_connectivity\_definition\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.464 Data\_element\_association to Functional\_unit\_allocation**

Each Data\_element\_association refers to exactly one Functional\_unit\_allocation in the role of associated\_item. Each Functional\_unit\_allocation acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.465 Data\_element\_association to Functionality**

Each Data\_element\_association refers to exactly one Functionality in the role of associated\_item. Each Functionality acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.466 Data\_element\_association to Generic\_note**

Each Data\_element\_association refers to exactly one Generic\_note in the role of associated\_item. Each Generic\_note acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.467 Data\_element\_association to Interface**

Each Data\_element\_association refers to exactly one Interface in the role of associated\_item. Each Interface acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.468 Data\_element\_association to Interface\_port**

Each Data\_element\_association refers to exactly one Interface\_port in the role of associated\_item. Each Interface\_port acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.469 Data\_element\_association to Interface\_terminal**

Each Data\_element\_association refers to exactly one Interface\_terminal in the role of associated\_item. Each Interface\_terminal acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.470 Data\_element\_association to Item**

Each Data\_element\_association refers to exactly one Item in the role of associated\_item. Each Item acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.471 Data\_element\_association to Item\_definition\_relationship**

Each Data\_element\_association refers to exactly one Item\_definition\_relationship in the role of associated\_item. Each Item\_definition\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.472 Data\_element\_association to Item\_version**

Each Data\_element\_association refers to exactly one Item\_version in the role of associated\_item. Each Item\_version acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.473 Data\_element\_association to Location**

Each Data\_element\_association refers to exactly one Location in the role of associated\_item. Each Location acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.474 Data\_element\_association to Location\_relationship**

Each Data\_element\_association refers to exactly one Location\_relationship in the role of associated\_item. Each Location\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.475 Data\_element\_association to Marking**

Each Data\_element\_association refers to exactly one Marking in the role of associated\_item. Each Marking acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.476 Data\_element\_association to Node**

Each Data\_element\_association refers to exactly one Node in the role of associated\_item. Each Node acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.477 Data\_element\_association to Node\_relationship**

Each Data\_element\_association refers to exactly one Node\_relationship in the role of associated\_item. Each Node\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.478 Data\_element\_association to Notification**

Each Data\_element\_association refers to exactly one Notification in the role of associated\_item. Each Notification acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.479 Data\_element\_association to Offered\_function\_allocation**

Each Data\_element\_association refers to exactly one Offered\_function\_allocation in the role of associated\_item. Each Offered\_function\_allocation acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.480 Data\_element\_association to Organization**

Each Data\_element\_association refers to zero or one Organization in the role of data\_element\_context. Each Organization acts as data\_element\_context for zero, one, or more Data\_element\_association objects.

#### **4.3.481 Data\_element\_association to Path**

Each Data\_element\_association refers to exactly one Path in the role of associated\_item. Each Path acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.482 Data\_element\_association to Path\_node**

Each Data\_element\_association refers to exactly one Path\_node in the role of associated\_item. Each Path\_node acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.483 Data\_element\_association to Physical\_assembly\_relationship**



Each Data\_element\_association refers to exactly one Physical\_assembly\_relationship in the role of associated\_item. Each Physical\_assembly\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.484 Data\_element\_association to Physical\_instance**

Each Data\_element\_association refers to exactly one Physical\_instance in the role of associated\_item. Each Physical\_instance acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.485 Data\_element\_association to Port**

Each Data\_element\_association refers to exactly one Port in the role of associated\_item. Each Port acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.486 Data\_element\_association to Port\_allocation**

Each Data\_element\_association refers to exactly one Port\_allocation in the role of associated\_item. Each Port\_allocation acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.487 Data\_element\_association to Port\_association**

Each Data\_element\_association refers to exactly one Port\_association in the role of associated\_item. Each Port\_association acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.488 Data\_element\_association to Preferred\_item\_allocation**

Each Data\_element\_association refers to exactly one Preferred\_item\_allocation in the role of associated\_item. Each Preferred\_item\_allocation acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.489 Data\_element\_association to Preferred\_item\_terminal\_allocation**

Each Data\_element\_association refers to exactly one Preferred\_item\_terminal\_allocation in the role of associated\_item. Each Preferred\_item\_terminal\_allocation acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.490 Data\_element\_association to Process\_variable**

Each Data\_element\_association refers to exactly one Process\_variable in the role of associated\_item. Each Process\_variable acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.491 Data\_element\_association to Product\_class**

Each Data\_element\_association refers to exactly one Product\_class in the role of associated\_item. Each Product\_class acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.492 Data\_element\_association to Product\_class**

Each Data\_element\_association refers to zero or one Product\_class in the role of data\_element\_-context. Each Product\_class acts as data\_element\_context for zero, one, or more Data\_element\_-association objects.

#### **4.3.493 Data\_element\_association to Product\_identification**

Each Data\_element\_association refers to exactly one Product\_identification in the role of associated\_item. Each Product\_identification acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.494 Data\_element\_association to Product\_structure\_relationship**

Each Data\_element\_association refers to exactly one Product\_structure\_relationship in the role of associated\_item. Each Product\_structure\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.495 Data\_element\_association to Requirement**

Each Data\_element\_association refers to exactly one Requirement in the role of associated\_item. Each Requirement acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.496 Data\_element\_association to Route**

Each Data\_element\_association refers to exactly one Route in the role of associated\_item. Each Route acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.497 Data\_element\_association to Route\_relationship**

Each Data\_element\_association refers to exactly one Route\_relationship in the role of associated\_item. Each Route\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.498 Data\_element\_association to Routed\_segment**

Each Data\_element\_association refers to exactly one Routed\_segment in the role of associated\_item. Each Routed\_segment acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.499 Data\_element\_association to Section**

Each Data\_element\_association refers to exactly one Section in the role of associated\_item. Each Section acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.500 Data\_element\_association to Section\_end**

Each Data\_element\_association refers to exactly one Section\_end in the role of associated\_item. Each Section\_end acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.501 Data\_element\_association to Section\_interface**

Each Data\_element\_association refers to exactly one Section\_interface in the role of associated\_item. Each Section\_interface acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.502 Data\_element\_association to Section\_interface\_relationship**

Each Data\_element\_association refers to exactly one Section\_interface\_relationship in the role of associated\_item. Each Section\_interface\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.503 Data\_element\_association to Section\_relationship**

Each Data\_element\_association refers to exactly one Section\_relationship in the role of associated\_item. Each Section\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.504 Data\_element\_association to Signal**

Each Data\_element\_association refers to exactly one Signal in the role of associated\_item. Each Signal acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.505 Data\_element\_association to Signal\_value**

Each Data\_element\_association refers to exactly one Signal\_value in the role of associated\_item. Each Signal\_value acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.506 Data\_element\_association to Technical\_system**

Each Data\_element\_association refers to exactly one Technical\_system in the role of associated\_item. Each Technical\_system acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.507 Data\_element\_association to Technical\_system**

Each Data\_element\_association refers to zero or one Technical\_system in the role of data\_element\_context. Each Technical\_system acts as data\_element\_context for zero, one, or more Data\_element\_association objects.

**4.3.508 Data\_element\_association to Technical\_system\_relationship**

Each Data\_element\_association refers to exactly one Technical\_system\_relationship in the role of associated\_item. Each Technical\_system\_relationship acts as associated\_item for zero, one, or more Data\_element\_association objects.

**4.3.509 Data\_element\_association to Terminal**

Each Data\_element\_association refers to exactly one Terminal in the role of associated\_item. Each Terminal acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.510 Data\_element\_association to Work\_order**

Each Data\_element\_association refers to exactly one Work\_order in the role of associated\_item. Each Work\_order acts as associated\_item for zero, one, or more Data\_element\_association objects.

#### **4.3.511 Data\_element\_definition to External\_library\_reference**

Each Data\_element\_definition refers to zero or one External\_library\_reference in the role of source. Each External\_library\_reference acts as source for zero, one, or more Data\_element\_definition objects.

#### **4.3.512 Data\_element\_definition to Property\_reference**

Each Data\_element\_definition refers to zero or one Property\_reference in the role of source. Each Property\_reference acts as source for zero, one, or more Data\_element\_definition objects.

#### **4.3.513 Data\_element\_definition\_relationship to Data\_element\_definition**

Each Data\_element\_definition\_relationship refers to exactly one Data\_element\_definition in the role of related. Each Data\_element\_definition acts as related for zero, one, or more Data\_element\_definition\_relationship objects.

#### **4.3.514 Data\_element\_definition\_relationship to Data\_element\_definition**

Each Data\_element\_definition\_relationship refers to exactly one Data\_element\_definition in the role of relating. Each Data\_element\_definition acts as relating for zero, one, or more Data\_element\_definition\_relationship objects.

#### **4.3.515 Data\_element\_relationship to Data\_element**

Each Data\_element\_relationship refers to exactly one Data\_element in the role of related. Each Data\_element acts as related for zero, one, or more Data\_element\_relationship objects.

#### **4.3.516 Data\_element\_relationship to Data\_element**

Each Data\_element\_relationship refers to exactly one Data\_element in the role of relating. Each Data\_element acts as relating for zero, one, or more Data\_element\_relationship objects.

#### **4.3.517 Data\_element\_specification to Data\_element\_definition**

Each Data\_element\_specification refers to exactly one Data\_element\_definition in the role of specification\_of. Each Data\_element\_definition acts as specification\_of for zero, one, or more Data\_element\_specification objects.

#### **4.3.518 Data\_element\_specification to Language**

Each Data\_element\_specification refers to zero or one Language in the role of language\_specification. Each Language acts as language\_specification for zero, one, or more Data\_element\_specification objects.

**4.3.519 Date\_and\_person\_assignment to Activity**

Each Date\_and\_person\_assignment is\_applied\_to one or more Activity objects. Each Activity is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.520 Date\_and\_person\_assignment to Activity\_element**

Each Date\_and\_person\_assignment is\_applied\_to one or more Activity\_element objects. Each Activity\_element is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.521 Date\_and\_person\_assignment to Activity\_method\_assignment**

Each Date\_and\_person\_assignment is\_applied\_to one or more Activity\_method\_assignment objects. Each Activity\_method\_assignment is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.522 Date\_and\_person\_assignment to Activity\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Activity\_relationship objects. Each Activity\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.523 Date\_and\_person\_assignment to Alternate\_item\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Alternate\_item\_relationship objects. Each Alternate\_item\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.524 Date\_and\_person\_assignment to Approval\_status**

Each Date\_and\_person\_assignment is\_applied\_to one or more Approval\_status objects. Each Approval\_status is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.525 Date\_and\_person\_assignment to Assembly\_component\_-relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Assembly\_component\_relationship objects. Each Assembly\_component\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.526 Date\_and\_person\_assignment to Assembly\_substitute\_-relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Assembly\_substitute\_relationship objects. Each Assembly\_substitute\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.527 Date\_and\_person\_assignment to Cable\_pull\_information**

Each Date\_and\_person\_assignment is\_applied\_to one or more Cable\_pull\_information objects. Each Cable\_pull\_information is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.528 Date\_and\_person\_assignment to Certification**

Each Date\_and\_person\_assignment is\_applied\_to one or more Certification objects. Each Certification is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.529 Date\_and\_person\_assignment to Class\_category\_association**

Each Date\_and\_person\_assignment is\_applied\_to one or more Class\_category\_association objects. Each Class\_category\_association is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.530 Date\_and\_person\_assignment to Class\_condition\_association**

Each Date\_and\_person\_assignment is\_applied\_to one or more Class\_condition\_association objects. Each Class\_condition\_association is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.531 Date\_and\_person\_assignment to Class\_inclusion\_association**

Each Date\_and\_person\_assignment is\_applied\_to one or more Class\_inclusion\_association objects. Each Class\_inclusion\_association is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.532 Date\_and\_person\_assignment to Class\_specification\_association**

Each Date\_and\_person\_assignment is\_applied\_to one or more Class\_specification\_association objects. Each Class\_specification\_association is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.533 Date\_and\_person\_assignment to Class\_structure\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Class\_structure\_relationship objects. Each Class\_structure\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.534 Date\_and\_person\_assignment to Classification\_association**

Each Date\_and\_person\_assignment is\_applied\_to one or more Classification\_association objects. Each Classification\_association is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.535 Date\_and\_person\_assignment to Classification\_attribute**

Each Date\_and\_person\_assignment is\_applied\_to one or more Classification\_attribute objects. Each Classification\_attribute is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.536 Date\_and\_person\_assignment to Classification\_system**

Each Date\_and\_person\_assignment is\_applied\_to one or more Classification\_system objects. Each Classification\_system is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.537 Date\_and\_person\_assignment to Complex\_product**

Each Date\_and\_person\_assignment is\_applied\_to one or more Complex\_product objects. Each Complex\_product is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.538 Date\_and\_person\_assignment to Complex\_product\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Complex\_product\_relationship objects. Each Complex\_product\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.539 Date\_and\_person\_assignment to Composition\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Composition\_relationship objects. Each Composition\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.540 Date\_and\_person\_assignment to Configuration**

Each Date\_and\_person\_assignment is\_applied\_to one or more Configuration objects. Each Configuration is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.541 Date\_and\_person\_assignment to Connectivity\_allocation**

Each Date\_and\_person\_assignment is\_applied\_to one or more Connectivity\_allocation objects. Each Connectivity\_allocation is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.542 Date\_and\_person\_assignment to Connectivity\_definition**

Each Date\_and\_person\_assignment is\_applied\_to one or more Connectivity\_definition objects. Each Connectivity\_definition is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.543 Date\_and\_person\_assignment to Connectivity\_definition\_-relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Connectivity\_definition\_relationship objects. Each Connectivity\_definition\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.544 Date\_and\_person\_assignment to Contract**

Each Date\_and\_person\_assignment is\_applied\_to one or more Contract objects. Each Contract is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.545 Date\_and\_person\_assignment to Data\_element**

Each Date\_and\_person\_assignment is\_applied\_to one or more Data\_element objects. Each Data\_element is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.546 Date\_and\_person\_assignment to Data\_element\_association**

Each Date\_and\_person\_assignment is\_applied\_to one or more Data\_element\_association objects. Each Data\_element\_association is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.547 Date\_and\_person\_assignment to Data\_element\_definition**

Each Date\_and\_person\_assignment is\_applied\_to one or more Data\_element\_definition objects. Each Data\_element\_definition is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.548 Date\_and\_person\_assignment to Data\_element\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Data\_element\_relationship objects. Each Data\_element\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.549 Date\_and\_person\_assignment to Data\_element\_specification**

Each Date\_and\_person\_assignment is\_applied\_to one or more Data\_element\_specification objects. Each Data\_element\_specification is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.550 Date\_and\_person\_assignment to Date\_and\_person\_or\_organization**

Each Date\_and\_person\_assignment refers to exactly one Date\_and\_person\_or\_organization in the role of assigned\_date\_and\_person. Each Date\_and\_person\_or\_organization acts as assigned\_date\_and\_person for zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.551 Date\_and\_person\_assignment to Design\_discipline\_item\_definition**

Each Date\_and\_person\_assignment is\_applied\_to one or more Design\_discipline\_item\_definition objects. Each Design\_discipline\_item\_definition is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.552 Date\_and\_person\_assignment to Device**

Each Date\_and\_person\_assignment is\_applied\_to one or more Device objects. Each Device is related to zero, one, or more Date\_and\_person\_assignment objects.



**4.3.553 Date\_and\_person\_assignment to Device\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Device\_relationship objects. Each Device\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.554 Date\_and\_person\_assignment to Document**

Each Date\_and\_person\_assignment is\_applied\_to one or more Document objects. Each Document is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.555 Date\_and\_person\_assignment to Document\_file**

Each Date\_and\_person\_assignment is\_applied\_to one or more Document\_file objects. Each Document\_file is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.556 Date\_and\_person\_assignment to Document\_file\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Document\_file\_relationship objects. Each Document\_file\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.557 Date\_and\_person\_assignment to Document\_representation**

Each Date\_and\_person\_assignment is\_applied\_to one or more Document\_representation objects. Each Document\_representation is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.558 Date\_and\_person\_assignment to Document\_version**

Each Date\_and\_person\_assignment is\_applied\_to one or more Document\_version objects. Each Document\_version is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.559 Date\_and\_person\_assignment to Document\_version\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Document\_version\_relationship objects. Each Document\_version\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.560 Date\_and\_person\_assignment to Drawing**

Each Date\_and\_person\_assignment is\_applied\_to one or more Drawing objects. Each Drawing is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.561 Date\_and\_person\_assignment to Drawing\_sequence**

Each Date\_and\_person\_assignment is\_applied\_to one or more Drawing\_sequence objects. Each Drawing\_sequence is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.562 Date\_and\_person\_assignment to Drawing\_sheet**

Each Date\_and\_person\_assignment is\_applied\_to one or more Drawing\_sheet objects. Each Drawing\_sheet is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.563 Date\_and\_person\_assignment to Drawing\_sheet\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Drawing\_sheet\_relationship objects. Each Drawing\_sheet\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.564 Date\_and\_person\_assignment to Free\_segment**

Each Date\_and\_person\_assignment is\_applied\_to one or more Free\_segment objects. Each Free\_segment is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.565 Date\_and\_person\_assignment to Function\_definition**

Each Date\_and\_person\_assignment is\_applied\_to one or more Function\_definition objects. Each Function\_definition is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.566 Date\_and\_person\_assignment to Function\_definition\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Function\_definition\_relationship objects. Each Function\_definition\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.567 Date\_and\_person\_assignment to Function\_interface**

Each Date\_and\_person\_assignment is\_applied\_to one or more Function\_interface objects. Each Function\_interface is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.568 Date\_and\_person\_assignment to Function\_unit**

Each Date\_and\_person\_assignment is\_applied\_to one or more Function\_unit objects. Each Function\_unit is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.569 Date\_and\_person\_assignment to Function\_unit\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Function\_unit\_relationship objects. Each Function\_unit\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.570 Date\_and\_person\_assignment to Function\_version**

Each Date\_and\_person\_assignment is\_applied\_to one or more Function\_version objects. Each Function\_version is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.571 Date\_and\_person\_assignment to Function\_version\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Function\_version\_relationship objects. Each Function\_version\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.572 Date\_and\_person\_assignment to Functional\_connectivity\_definition**

Each Date\_and\_person\_assignment is\_applied\_to one or more Functional\_connectivity\_definition objects. Each Functional\_connectivity\_definition is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.573 Date\_and\_person\_assignment to Functional\_connectivity\_definition\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Functional\_connectivity\_definition\_relationship objects. Each Functional\_connectivity\_definition\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.574 Date\_and\_person\_assignment to Functional\_unit\_allocation**

Each Date\_and\_person\_assignment is\_applied\_to one or more Functional\_unit\_allocation objects. Each Functional\_unit\_allocation is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.575 Date\_and\_person\_assignment to Functionality**

Each Date\_and\_person\_assignment is\_applied\_to one or more Functionality objects. Each Functionality is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.576 Date\_and\_person\_assignment to General\_classification**

Each Date\_and\_person\_assignment is\_applied\_to one or more General\_classification objects. Each General\_classification is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.577 Date\_and\_person\_assignment to Generic\_note**

Each Date\_and\_person\_assignment is\_applied\_to one or more Generic\_note objects. Each Generic\_note is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.578 Date\_and\_person\_assignment to Interface**

Each Date\_and\_person\_assignment is\_applied\_to one or more Interface objects. Each Interface is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.579 Date\_and\_person\_assignment to Interface\_port**

Each Date\_and\_person\_assignment is\_applied\_to one or more Interface\_port objects. Each Interface\_port is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.580 Date\_and\_person\_assignment to Interface\_terminal**

Each Date\_and\_person\_assignment is\_applied\_to one or more Interface\_terminal objects. Each Interface\_terminal is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.581 Date\_and\_person\_assignment to Item**

Each Date\_and\_person\_assignment is\_applied\_to one or more Item objects. Each Item is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.582 Date\_and\_person\_assignment to Item\_definition\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Item\_definition\_relationship objects. Each Item\_definition\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.583 Date\_and\_person\_assignment to Item\_version**

Each Date\_and\_person\_assignment is\_applied\_to one or more Item\_version objects. Each Item\_version is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.584 Date\_and\_person\_assignment to Item\_version\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Item\_version\_relationship objects. Each Item\_version\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.585 Date\_and\_person\_assignment to Location**

Each Date\_and\_person\_assignment is\_applied\_to one or more Location objects. Each Location is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.586 Date\_and\_person\_assignment to Location\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Location\_relationship objects. Each Location\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.587 Date\_and\_person\_assignment to Marking**

Each Date\_and\_person\_assignment is\_applied\_to one or more Marking objects. Each Marking is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.588 Date\_and\_person\_assignment to Material**

Each Date\_and\_person\_assignment is\_applied\_to one or more Material objects. Each Material is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.589 Date\_and\_person\_assignment to Node**

Each Date\_and\_person\_assignment is\_applied\_to one or more Node objects. Each Node is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.590 Date\_and\_person\_assignment to Node\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Node\_relationship objects. Each Node\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.591 Date\_and\_person\_assignment to Notification**

Each Date\_and\_person\_assignment is\_applied\_to one or more Notification objects. Each Notification is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.592 Date\_and\_person\_assignment to Notification\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Notification\_relationship objects. Each Notification\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.593 Date\_and\_person\_assignment to Offered\_function\_allocation**

Each Date\_and\_person\_assignment is\_applied\_to one or more Offered\_function\_allocation objects. Each Offered\_function\_allocation is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.594 Date\_and\_person\_assignment to Organization\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Organization\_relationship objects. Each Organization\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.595 Date\_and\_person\_assignment to Path**

Each Date\_and\_person\_assignment is\_applied\_to one or more Path objects. Each Path is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.596 Date\_and\_person\_assignment to Path\_node**

Each Date\_and\_person\_assignment is\_applied\_to one or more Path\_node objects. Each Path\_node is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.597 Date\_and\_person\_assignment to Path\_node\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Path\_node\_relationship objects. Each Path\_node\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.598 Date\_and\_person\_assignment to Path\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Path\_relationship objects. Each Path\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.599 Date\_and\_person\_assignment to Person\_in\_organization**

Each Date\_and\_person\_assignment is\_applied\_to one or more Person\_in\_organization objects. Each Person\_in\_organization is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.600 Date\_and\_person\_assignment to Person\_in\_organization - relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Person\_in\_organization\_relationship objects. Each Person\_in\_organization\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.601 Date\_and\_person\_assignment to Physical\_assembly\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Physical\_assembly\_relationship objects. Each Physical\_assembly\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.602 Date\_and\_person\_assignment to Physical\_instance**

Each Date\_and\_person\_assignment is\_applied\_to one or more Physical\_instance objects. Each Physical\_instance is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.603 Date\_and\_person\_assignment to Port**

Each Date\_and\_person\_assignment is\_applied\_to one or more Port objects. Each Port is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.604 Date\_and\_person\_assignment to Port\_allocation**

Each Date\_and\_person\_assignment is\_applied\_to one or more Port\_allocation objects. Each Port\_allocation is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.605 Date\_and\_person\_assignment to Port\_association**

Each Date\_and\_person\_assignment is\_applied\_to one or more Port\_association objects. Each Port\_association is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.606 Date\_and\_person\_assignment to Preferred\_item\_allocation**

Each Date\_and\_person\_assignment is\_applied\_to one or more Preferred\_item\_allocation objects. Each Preferred\_item\_allocation is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.607 Date\_and\_person\_assignment to Preferred\_item\_terminal\_allocation**

Each Date\_and\_person\_assignment is\_applied\_to one or more Preferred\_item\_terminal\_allocation objects. Each Preferred\_item\_terminal\_allocation is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.608 Date\_and\_person\_assignment to Process\_variable**

Each Date\_and\_person\_assignment is\_applied\_to one or more Process\_variable objects. Each Process\_variable is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.609 Date\_and\_person\_assignment to Process\_variable\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Process\_variable\_relationship objects. Each Process\_variable\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.610 Date\_and\_person\_assignment to Product\_class**

Each Date\_and\_person\_assignment is\_applied\_to one or more Product\_class objects. Each Product\_class is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.611 Date\_and\_person\_assignment to Product\_identification**

Each Date\_and\_person\_assignment is\_applied\_to one or more Product\_identification objects. Each Product\_identification is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.612 Date\_and\_person\_assignment to Product\_structure\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Product\_structure\_relationship objects. Each Product\_structure\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.613 Date\_and\_person\_assignment to Project**

Each Date\_and\_person\_assignment is\_applied\_to one or more Project objects. Each Project is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.614 Date\_and\_person\_assignment to Requirement**

Each Date\_and\_person\_assignment is\_applied\_to one or more Requirement objects. Each Requirement is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.615 Date\_and\_person\_assignment to Requirement\_document\_assignment**

Each Date\_and\_person\_assignment is\_applied\_to one or more Requirement\_document\_assignment objects. Each Requirement\_document\_assignment is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.616 Date\_and\_person\_assignment to Route**

Each Date\_and\_person\_assignment is\_applied\_to one or more Route objects. Each Route is related to zero, one, or more Date\_and\_person\_assignment objects.



#### **4.3.617 Date\_and\_person\_assignment to Route\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Route\_relationship objects. Each Route\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.618 Date\_and\_person\_assignment to Routed\_segment**

Each Date\_and\_person\_assignment is\_applied\_to one or more Routed\_segment objects. Each Routed\_segment is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.619 Date\_and\_person\_assignment to Section**

Each Date\_and\_person\_assignment is\_applied\_to one or more Section objects. Each Section is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.620 Date\_and\_person\_assignment to Section\_end**

Each Date\_and\_person\_assignment is\_applied\_to one or more Section\_end objects. Each Section\_end is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.621 Date\_and\_person\_assignment to Section\_interface**

Each Date\_and\_person\_assignment is\_applied\_to one or more Section\_interface objects. Each Section\_interface is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.622 Date\_and\_person\_assignment to Section\_interface\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Section\_interface\_relationship objects. Each Section\_interface\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.623 Date\_and\_person\_assignment to Section\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Section\_relationship objects. Each Section\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.624 Date\_and\_person\_assignment to Security\_classification**

Each Date\_and\_person\_assignment is\_applied\_to one or more Security\_classification objects. Each Security\_classification is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.625 Date\_and\_person\_assignment to Security\_level**

Each Date\_and\_person\_assignment is\_applied\_to one or more Security\_level objects. Each Security\_level is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.626 Date\_and\_person\_assignment to Signal**

Each Date\_and\_person\_assignment is\_applied\_to one or more Signal objects. Each Signal is related to zero, one, or more Date\_and\_person\_assignment objects.



**4.3.627 Date\_and\_person\_assignment to Signal\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Signal\_relationship objects. Each Signal\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.628 Date\_and\_person\_assignment to Signal\_value**

Each Date\_and\_person\_assignment is\_applied\_to one or more Signal\_value objects. Each Signal\_value is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.629 Date\_and\_person\_assignment to Specification**

Each Date\_and\_person\_assignment is\_applied\_to one or more Specification objects. Each Specification is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.630 Date\_and\_person\_assignment to Specification\_category**

Each Date\_and\_person\_assignment is\_applied\_to one or more Specification\_category objects. Each Specification\_category is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.631 Date\_and\_person\_assignment to Specification\_expression**

Each Date\_and\_person\_assignment is\_applied\_to one or more Specification\_expression objects. Each Specification\_expression is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.632 Date\_and\_person\_assignment to Specification\_inclusion**

Each Date\_and\_person\_assignment is\_applied\_to one or more Specification\_inclusion objects. Each Specification\_inclusion is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.633 Date\_and\_person\_assignment to Technical\_system**

Each Date\_and\_person\_assignment is\_applied\_to one or more Technical\_system objects. Each Technical\_system is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.634 Date\_and\_person\_assignment to Technical\_system\_relationship**

Each Date\_and\_person\_assignment is\_applied\_to one or more Technical\_system\_relationship objects. Each Technical\_system\_relationship is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.635 Date\_and\_person\_assignment to Terminal**

Each Date\_and\_person\_assignment is\_applied\_to one or more Terminal objects. Each Terminal is related to zero, one, or more Date\_and\_person\_assignment objects.

**4.3.636 Date\_and\_person\_assignment to Work\_order**

Each Date\_and\_person\_assignment is\_applied\_to one or more Work\_order objects. Each Work\_order is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.637 Date\_and\_person\_assignment to Work\_request**

Each Date\_and\_person\_assignment is\_applied\_to one or more Work\_request objects. Each Work\_request is related to zero, one, or more Date\_and\_person\_assignment objects.

#### **4.3.638 Date\_and\_person\_or\_organization to Date\_time**

Each Date\_and\_person\_or\_organization refers to exactly one Date\_time in the role of associated\_date. Each Date\_time acts as associated\_date for zero, one, or more Date\_and\_person\_or\_organization objects.

#### **4.3.639 Date\_and\_person\_or\_organization to Organization**

Each Date\_and\_person\_or\_organization refers to exactly one Organization in the role of person\_or\_organization. Each Organization acts as person\_or\_organization for zero, one, or more Date\_and\_person\_or\_organization objects.

#### **4.3.640 Date\_and\_person\_or\_organization to Person\_in\_organization**

Each Date\_and\_person\_or\_organization refers to exactly one Person\_in\_organization in the role of person\_or\_organization. Each Person\_in\_organization acts as person\_or\_organization for zero, one, or more Date\_and\_person\_or\_organization objects.

#### **4.3.641 Date\_time\_assignment to Activity**

Each Date\_time\_assignment is\_applied\_to one or more Activity objects. Each Activity is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.642 Date\_time\_assignment to Activity\_element**

Each Date\_time\_assignment is\_applied\_to one or more Activity\_element objects. Each Activity\_element is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.643 Date\_time\_assignment to Activity\_method\_assignment**

Each Date\_time\_assignment is\_applied\_to one or more Activity\_method\_assignment objects. Each Activity\_method\_assignment is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.644 Date\_time\_assignment to Activity\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Activity\_relationship objects. Each Activity\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.645 Date\_time\_assignment to Alternate\_item\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Alternate\_item\_relationship objects. Each Alternate\_item\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.646 Date\_time\_assignment to Approval\_status**

Each Date\_time\_assignment is\_applied\_to one or more Approval\_status objects. Each Approval\_status is related to zero, one, or more Date\_time\_assignment objects.

**4.3.647 Date\_time\_assignment to Assembly\_component\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Assembly\_component\_relationship objects. Each Assembly\_component\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.648 Date\_time\_assignment to Assembly\_substitute\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Assembly\_substitute\_relationship objects. Each Assembly\_substitute\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.649 Date\_time\_assignment to Cable\_pull\_information**

Each Date\_time\_assignment is\_applied\_to one or more Cable\_pull\_information objects. Each Cable\_pull\_information is related to zero, one, or more Date\_time\_assignment objects.

**4.3.650 Date\_time\_assignment to Certification**

Each Date\_time\_assignment is\_applied\_to one or more Certification objects. Each Certification is related to zero, one, or more Date\_time\_assignment objects.

**4.3.651 Date\_time\_assignment to Class\_category\_association**

Each Date\_time\_assignment is\_applied\_to one or more Class\_category\_association objects. Each Class\_category\_association is related to zero, one, or more Date\_time\_assignment objects.

**4.3.652 Date\_time\_assignment to Class\_condition\_association**

Each Date\_time\_assignment is\_applied\_to one or more Class\_condition\_association objects. Each Class\_condition\_association is related to zero, one, or more Date\_time\_assignment objects.

**4.3.653 Date\_time\_assignment to Class\_inclusion\_association**

Each Date\_time\_assignment is\_applied\_to one or more Class\_inclusion\_association objects. Each Class\_inclusion\_association is related to zero, one, or more Date\_time\_assignment objects.

**4.3.654 Date\_time\_assignment to Class\_specification\_association**

Each Date\_time\_assignment is\_applied\_to one or more Class\_specification\_association objects. Each Class\_specification\_association is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.655 Date\_time\_assignment to Class\_structure\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Class\_structure\_relationship objects. Each Class\_structure\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.656 Date\_time\_assignment to Classification\_association**

Each Date\_time\_assignment is\_applied\_to one or more Classification\_association objects. Each Classification\_association is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.657 Date\_time\_assignment to Classification\_attribute**

Each Date\_time\_assignment is\_applied\_to one or more Classification\_attribute objects. Each Classification\_attribute is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.658 Date\_time\_assignment to Classification\_system**

Each Date\_time\_assignment is\_applied\_to one or more Classification\_system objects. Each Classification\_system is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.659 Date\_time\_assignment to Complex\_product**

Each Date\_time\_assignment is\_applied\_to one or more Complex\_product objects. Each Complex\_product is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.660 Date\_time\_assignment to Complex\_product\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Complex\_product\_relationship objects. Each Complex\_product\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.661 Date\_time\_assignment to Composition\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Composition\_relationship objects. Each Composition\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.662 Date\_time\_assignment to Configuration**

Each Date\_time\_assignment is\_applied\_to one or more Configuration objects. Each Configuration is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.663 Date\_time\_assignment to Connectivity\_allocation**

Each Date\_time\_assignment is\_applied\_to one or more Connectivity\_allocation objects. Each Connectivity\_allocation is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.664 Date\_time\_assignment to Connectivity\_definition**

Each Date\_time\_assignment is\_applied\_to one or more Connectivity\_definition objects. Each Connectivity\_definition is related to zero, one, or more Date\_time\_assignment objects.

**4.3.665 Date\_time\_assignment to Connectivity\_definition\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Connectivity\_definition\_relationship objects. Each Connectivity\_definition\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.666 Date\_time\_assignment to Contract**

Each Date\_time\_assignment is\_applied\_to one or more Contract objects. Each Contract is related to zero, one, or more Date\_time\_assignment objects.

**4.3.667 Date\_time\_assignment to Data\_element**

Each Date\_time\_assignment is\_applied\_to one or more Data\_element objects. Each Data\_element is related to zero, one, or more Date\_time\_assignment objects.

**4.3.668 Date\_time\_assignment to Data\_element\_association**

Each Date\_time\_assignment is\_applied\_to one or more Data\_element\_association objects. Each Data\_element\_association is related to zero, one, or more Date\_time\_assignment objects.

**4.3.669 Date\_time\_assignment to Data\_element\_definition**

Each Date\_time\_assignment is\_applied\_to one or more Data\_element\_definition objects. Each Data\_element\_definition is related to zero, one, or more Date\_time\_assignment objects.

**4.3.670 Date\_time\_assignment to Data\_element\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Data\_element\_relationship objects. Each Data\_element\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.671 Date\_time\_assignment to Data\_element\_specification**

Each Date\_time\_assignment is\_applied\_to one or more Data\_element\_specification objects. Each Data\_element\_specification is related to zero, one, or more Date\_time\_assignment objects.

**4.3.672 Date\_time\_assignment to Date\_time**

Each Date\_time\_assignment refers to exactly one Date\_time in the role of assigned\_date\_time. Each Date\_time acts as assigned\_date\_time for zero, one, or more Date\_time\_assignment objects.

**4.3.673 Date\_time\_assignment to Design\_discipline\_item\_definition**

Each Date\_time\_assignment is\_applied\_to one or more Design\_discipline\_item\_definition objects. Each Design\_discipline\_item\_definition is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.674 Date\_time\_assignment to Device**

Each Date\_time\_assignment is\_applied\_to one or more Device objects. Each Device is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.675 Date\_time\_assignment to Device\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Device\_relationship objects. Each Device\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.676 Date\_time\_assignment to Document**

Each Date\_time\_assignment is\_applied\_to one or more Document objects. Each Document is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.677 Date\_time\_assignment to Document\_file**

Each Date\_time\_assignment is\_applied\_to one or more Document\_file objects. Each Document\_file is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.678 Date\_time\_assignment to Document\_file\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Document\_file\_relationship objects. Each Document\_file\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.679 Date\_time\_assignment to Document\_representation**

Each Date\_time\_assignment is\_applied\_to one or more Document\_representation objects. Each Document\_representation is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.680 Date\_time\_assignment to Document\_version**

Each Date\_time\_assignment is\_applied\_to one or more Document\_version objects. Each Document\_version is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.681 Date\_time\_assignment to Document\_version\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Document\_version\_relationship objects. Each Document\_version\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.682 Date\_time\_assignment to Drawing**

Each Date\_time\_assignment is\_applied\_to one or more Drawing objects. Each Drawing is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.683 Date\_time\_assignment to Drawing\_sequence**

Each Date\_time\_assignment is\_applied\_to one or more Drawing\_sequence objects. Each Drawing\_sequence is related to zero, one, or more Date\_time\_assignment objects.

**4.3.684 Date\_time\_assignment to Drawing\_sheet**

Each Date\_time\_assignment is\_applied\_to one or more Drawing\_sheet objects. Each Drawing\_sheet is related to zero, one, or more Date\_time\_assignment objects.

**4.3.685 Date\_time\_assignment to Drawing\_sheet\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Drawing\_sheet\_relationship objects. Each Drawing\_sheet\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.686 Date\_time\_assignment to Free\_segment**

Each Date\_time\_assignment is\_applied\_to one or more Free\_segment objects. Each Free\_segment is related to zero, one, or more Date\_time\_assignment objects.

**4.3.687 Date\_time\_assignment to Function\_definition**

Each Date\_time\_assignment is\_applied\_to one or more Function\_definition objects. Each Function\_definition is related to zero, one, or more Date\_time\_assignment objects.

**4.3.688 Date\_time\_assignment to Function\_definition\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Function\_definition\_relationship objects. Each Function\_definition\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.689 Date\_time\_assignment to Function\_interface**

Each Date\_time\_assignment is\_applied\_to one or more Function\_interface objects. Each Function\_interface is related to zero, one, or more Date\_time\_assignment objects.

**4.3.690 Date\_time\_assignment to Function\_unit**

Each Date\_time\_assignment is\_applied\_to one or more Function\_unit objects. Each Function\_unit is related to zero, one, or more Date\_time\_assignment objects.

**4.3.691 Date\_time\_assignment to Function\_unit\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Function\_unit\_relationship objects. Each Function\_unit\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.692 Date\_time\_assignment to Function\_version**

Each Date\_time\_assignment is\_applied\_to one or more Function\_version objects. Each Function\_version is related to zero, one, or more Date\_time\_assignment objects.

**4.3.693 Date\_time\_assignment to Function\_version\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Function\_version\_relationship objects. Each Function\_version\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.694 Date\_time\_assignment to Functional\_connectivity\_definition**

Each Date\_time\_assignment is\_applied\_to one or more Functional\_connectivity\_definition objects. Each Functional\_connectivity\_definition is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.695 Date\_time\_assignment to Functional\_connectivity\_definition - relationship**

Each Date\_time\_assignment is\_applied\_to one or more Functional\_connectivity\_definition - relationship objects. Each Functional\_connectivity\_definition\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.696 Date\_time\_assignment to Functional\_unit\_allocation**

Each Date\_time\_assignment is\_applied\_to one or more Functional\_unit\_allocation objects. Each Functional\_unit\_allocation is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.697 Date\_time\_assignment to Functionality**

Each Date\_time\_assignment is\_applied\_to one or more Functionality objects. Each Functionality is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.698 Date\_time\_assignment to General\_classification**

Each Date\_time\_assignment is\_applied\_to one or more General\_classification objects. Each General\_classification is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.699 Date\_time\_assignment to Generic\_note**

Each Date\_time\_assignment is\_applied\_to one or more Generic\_note objects. Each Generic\_note is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.700 Date\_time\_assignment to Interface**

Each Date\_time\_assignment is\_applied\_to one or more Interface objects. Each Interface is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.701 Date\_time\_assignment to Interface\_port**

Each Date\_time\_assignment is\_applied\_to one or more Interface\_port objects. Each Interface\_port is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.702 Date\_time\_assignment to Interface\_terminal**

Each Date\_time\_assignment is\_applied\_to one or more Interface\_terminal objects. Each Interface\_terminal is related to zero, one, or more Date\_time\_assignment objects.



**4.3.703 Date\_time\_assignment to Item**

Each Date\_time\_assignment is\_applied\_to one or more Item objects. Each Item is related to zero, one, or more Date\_time\_assignment objects.

**4.3.704 Date\_time\_assignment to Item\_definition\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Item\_definition\_relationship objects. Each Item\_definition\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.705 Date\_time\_assignment to Item\_version**

Each Date\_time\_assignment is\_applied\_to one or more Item\_version objects. Each Item\_version is related to zero, one, or more Date\_time\_assignment objects.

**4.3.706 Date\_time\_assignment to Item\_version\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Item\_version\_relationship objects. Each Item\_version\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.707 Date\_time\_assignment to Location**

Each Date\_time\_assignment is\_applied\_to one or more Location objects. Each Location is related to zero, one, or more Date\_time\_assignment objects.

**4.3.708 Date\_time\_assignment to Location\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Location\_relationship objects. Each Location\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.709 Date\_time\_assignment to Marking**

Each Date\_time\_assignment is\_applied\_to one or more Marking objects. Each Marking is related to zero, one, or more Date\_time\_assignment objects.

**4.3.710 Date\_time\_assignment to Material**

Each Date\_time\_assignment is\_applied\_to one or more Material objects. Each Material is related to zero, one, or more Date\_time\_assignment objects.

**4.3.711 Date\_time\_assignment to Node**

Each Date\_time\_assignment is\_applied\_to one or more Node objects. Each Node is related to zero, one, or more Date\_time\_assignment objects.

**4.3.712 Date\_time\_assignment to Node\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Node\_relationship objects. Each Node\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.713 Date\_time\_assignment to Notification**

Each Date\_time\_assignment is\_applied\_to one or more Notification objects. Each Notification is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.714 Date\_time\_assignment to Notification\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Notification\_relationship objects. Each Notification\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.715 Date\_time\_assignment to Offered\_function\_allocation**

Each Date\_time\_assignment is\_applied\_to one or more Offered\_function\_allocation objects. Each Offered\_function\_allocation is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.716 Date\_time\_assignment to Organization\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Organization\_relationship objects. Each Organization\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.717 Date\_time\_assignment to Path**

Each Date\_time\_assignment is\_applied\_to one or more Path objects. Each Path is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.718 Date\_time\_assignment to Path\_node**

Each Date\_time\_assignment is\_applied\_to one or more Path\_node objects. Each Path\_node is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.719 Date\_time\_assignment to Path\_node\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Path\_node\_relationship objects. Each Path\_node\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.720 Date\_time\_assignment to Path\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Path\_relationship objects. Each Path\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.721 Date\_time\_assignment to Person\_in\_organization**

Each Date\_time\_assignment is\_applied\_to one or more Person\_in\_organization objects. Each Person\_in\_organization is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.722 Date\_time\_assignment to Person\_in\_organization\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Person\_in\_organization\_relationship objects. Each Person\_in\_organization\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.723 Date\_time\_assignment to Physical\_assembly\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Physical\_assembly\_relationship objects. Each Physical\_assembly\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.724 Date\_time\_assignment to Physical\_instance**

Each Date\_time\_assignment is\_applied\_to one or more Physical\_instance objects. Each Physical\_instance is related to zero, one, or more Date\_time\_assignment objects.

**4.3.725 Date\_time\_assignment to Port**

Each Date\_time\_assignment is\_applied\_to one or more Port objects. Each Port is related to zero, one, or more Date\_time\_assignment objects.

**4.3.726 Date\_time\_assignment to Port\_allocation**

Each Date\_time\_assignment is\_applied\_to one or more Port\_allocation objects. Each Port\_allocation is related to zero, one, or more Date\_time\_assignment objects.

**4.3.727 Date\_time\_assignment to Port\_association**

Each Date\_time\_assignment is\_applied\_to one or more Port\_association objects. Each Port\_association is related to zero, one, or more Date\_time\_assignment objects.

**4.3.728 Date\_time\_assignment to Preferred\_item\_allocation**

Each Date\_time\_assignment is\_applied\_to one or more Preferred\_item\_allocation objects. Each Preferred\_item\_allocation is related to zero, one, or more Date\_time\_assignment objects.

**4.3.729 Date\_time\_assignment to Preferred\_item\_terminal\_allocation**

Each Date\_time\_assignment is\_applied\_to one or more Preferred\_item\_terminal\_allocation objects. Each Preferred\_item\_terminal\_allocation is related to zero, one, or more Date\_time\_assignment objects.

**4.3.730 Date\_time\_assignment to Process\_variable**

Each Date\_time\_assignment is\_applied\_to one or more Process\_variable objects. Each Process\_variable is related to zero, one, or more Date\_time\_assignment objects.

**4.3.731 Date\_time\_assignment to Process\_variable\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Process\_variable\_relationship objects. Each Process\_variable\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.732 Date\_time\_assignment to Product\_class**

Each Date\_time\_assignment is\_applied\_to one or more Product\_class objects. Each Product\_class is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.733 Date\_time\_assignment to Product\_identification**

Each Date\_time\_assignment is\_applied\_to one or more Product\_identification objects. Each Product\_identification is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.734 Date\_time\_assignment to Product\_structure\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Product\_structure\_relationship objects. Each Product\_structure\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.735 Date\_time\_assignment to Project**

Each Date\_time\_assignment is\_applied\_to one or more Project objects. Each Project is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.736 Date\_time\_assignment to Requirement**

Each Date\_time\_assignment is\_applied\_to one or more Requirement objects. Each Requirement is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.737 Date\_time\_assignment to Requirement\_document\_assignment**

Each Date\_time\_assignment is\_applied\_to one or more Requirement\_document\_assignment objects. Each Requirement\_document\_assignment is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.738 Date\_time\_assignment to Route**

Each Date\_time\_assignment is\_applied\_to one or more Route objects. Each Route is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.739 Date\_time\_assignment to Route\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Route\_relationship objects. Each Route\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.740 Date\_time\_assignment to Routed\_segment**

Each Date\_time\_assignment is\_applied\_to one or more Routed\_segment objects. Each Routed\_segment is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.741 Date\_time\_assignment to Section**

Each Date\_time\_assignment is\_applied\_to one or more Section objects. Each Section is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.742 Date\_time\_assignment to Section\_end**

Each Date\_time\_assignment is\_applied\_to one or more Section\_end objects. Each Section\_end is related to zero, one, or more Date\_time\_assignment objects.

**4.3.743 Date\_time\_assignment to Section\_interface**

Each Date\_time\_assignment is\_applied\_to one or more Section\_interface objects. Each Section\_interface is related to zero, one, or more Date\_time\_assignment objects.

**4.3.744 Date\_time\_assignment to Section\_interface\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Section\_interface\_relationship objects. Each Section\_interface\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.745 Date\_time\_assignment to Section\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Section\_relationship objects. Each Section\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.746 Date\_time\_assignment to Security\_classification**

Each Date\_time\_assignment is\_applied\_to one or more Security\_classification objects. Each Security\_classification is related to zero, one, or more Date\_time\_assignment objects.

**4.3.747 Date\_time\_assignment to Security\_level**

Each Date\_time\_assignment is\_applied\_to one or more Security\_level objects. Each Security\_level is related to zero, one, or more Date\_time\_assignment objects.

**4.3.748 Date\_time\_assignment to Signal**

Each Date\_time\_assignment is\_applied\_to one or more Signal objects. Each Signal is related to zero, one, or more Date\_time\_assignment objects.

**4.3.749 Date\_time\_assignment to Signal\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Signal\_relationship objects. Each Signal\_relationship is related to zero, one, or more Date\_time\_assignment objects.

**4.3.750 Date\_time\_assignment to Signal\_value**

Each Date\_time\_assignment is\_applied\_to one or more Signal\_value objects. Each Signal\_value is related to zero, one, or more Date\_time\_assignment objects.

**4.3.751 Date\_time\_assignment to Specification**

Each Date\_time\_assignment is\_applied\_to one or more Specification objects. Each Specification is related to zero, one, or more Date\_time\_assignment objects.

**4.3.752 Date\_time\_assignment to Specification\_category**

Each Date\_time\_assignment is\_applied\_to one or more Specification\_category objects. Each Specification\_category is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.753 Date\_time\_assignment to Specification\_expression**

Each Date\_time\_assignment is\_applied\_to one or more Specification\_expression objects. Each Specification\_expression is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.754 Date\_time\_assignment to Specification\_inclusion**

Each Date\_time\_assignment is\_applied\_to one or more Specification\_inclusion objects. Each Specification\_inclusion is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.755 Date\_time\_assignment to Technical\_system**

Each Date\_time\_assignment is\_applied\_to one or more Technical\_system objects. Each Technical\_system is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.756 Date\_time\_assignment to Technical\_system\_relationship**

Each Date\_time\_assignment is\_applied\_to one or more Technical\_system\_relationship objects. Each Technical\_system\_relationship is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.757 Date\_time\_assignment to Terminal**

Each Date\_time\_assignment is\_applied\_to one or more Terminal objects. Each Terminal is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.758 Date\_time\_assignment to Work\_order**

Each Date\_time\_assignment is\_applied\_to one or more Work\_order objects. Each Work\_order is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.759 Date\_time\_assignment to Work\_request**

Each Date\_time\_assignment is\_applied\_to one or more Work\_request objects. Each Work\_request is related to zero, one, or more Date\_time\_assignment objects.

#### **4.3.760 Date\_time\_interval\_assignment to Activity**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Activity objects. Each Activity is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.761 Date\_time\_interval\_assignment to Activity\_element**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Activity\_element objects. Each Activity\_element is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.762 Date\_time\_interval\_assignment to Activity\_method\_assignment**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Activity\_method\_assignment objects. Each Activity\_method\_assignment is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.763 Date\_time\_interval\_assignment to Activity\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Activity\_relationship objects. Each Activity\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.764 Date\_time\_interval\_assignment to Alternate\_item\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Alternate\_item\_relationship objects. Each Alternate\_item\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.765 Date\_time\_interval\_assignment to Approval\_status**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Approval\_status objects. Each Approval\_status is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.766 Date\_time\_interval\_assignment to Assembly\_component\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Assembly\_component\_relationship objects. Each Assembly\_component\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.767 Date\_time\_interval\_assignment to Assembly\_substitute\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Assembly\_substitute\_relationship objects. Each Assembly\_substitute\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.768 Date\_time\_interval\_assignment to Cable\_pull\_information**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Cable\_pull\_information objects. Each Cable\_pull\_information is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.769 Date\_time\_interval\_assignment to Certification**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Certification objects. Each Certification is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.770 Date\_time\_interval\_assignment to Class\_category\_association**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Class\_category\_association objects. Each Class\_category\_association is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.771 Date\_time\_interval\_assignment to Class\_condition\_association**

Each Date\_time\_interval\_assignment is applied to one or more Class\_condition\_association objects. Each Class\_condition\_association is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.772 Date\_time\_interval\_assignment to Class\_inclusion\_association**

Each Date\_time\_interval\_assignment is applied to one or more Class\_inclusion\_association objects. Each Class\_inclusion\_association is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.773 Date\_time\_interval\_assignment to Class\_specification\_association**

Each Date\_time\_interval\_assignment is applied to one or more Class\_specification\_association objects. Each Class\_specification\_association is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.774 Date\_time\_interval\_assignment to Class\_structure\_relationship**

Each Date\_time\_interval\_assignment is applied to one or more Class\_structure\_relationship objects. Each Class\_structure\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.775 Date\_time\_interval\_assignment to Classification\_association**

Each Date\_time\_interval\_assignment is applied to one or more Classification\_association objects. Each Classification\_association is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.776 Date\_time\_interval\_assignment to Classification\_attribute**

Each Date\_time\_interval\_assignment is applied to one or more Classification\_attribute objects. Each Classification\_attribute is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.777 Date\_time\_interval\_assignment to Classification\_system**

Each Date\_time\_interval\_assignment is applied to one or more Classification\_system objects. Each Classification\_system is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.778 Date\_time\_interval\_assignment to Complex\_product**

Each Date\_time\_interval\_assignment is applied to one or more Complex\_product objects. Each Complex\_product is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.779 Date\_time\_interval\_assignment to Complex\_product\_relationship**

Each Date\_time\_interval\_assignment is applied to one or more Complex\_product\_relationship objects. Each Complex\_product\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.



**4.3.780 Date\_time\_interval\_assignment to Composition\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Composition\_relationship objects. Each Composition\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.781 Date\_time\_interval\_assignment to Configuration**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Configuration objects. Each Configuration is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.782 Date\_time\_interval\_assignment to Connectivity\_allocation**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Connectivity\_allocation objects. Each Connectivity\_allocation is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.783 Date\_time\_interval\_assignment to Connectivity\_definition**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Connectivity\_definition objects. Each Connectivity\_definition is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.784 Date\_time\_interval\_assignment to Connectivity\_definition\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Connectivity\_definition\_relationship objects. Each Connectivity\_definition\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.785 Date\_time\_interval\_assignment to Contract**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Contract objects. Each Contract is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.786 Date\_time\_interval\_assignment to Data\_element**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Data\_element objects. Each Data\_element is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.787 Date\_time\_interval\_assignment to Data\_element\_association**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Data\_element\_association objects. Each Data\_element\_association is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.788 Date\_time\_interval\_assignment to Data\_element\_definition**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Data\_element\_definition objects. Each Data\_element\_definition is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.789 Date\_time\_interval\_assignment to Data\_element\_relationship**

Each Date\_time\_interval\_assignment is applied to one or more Data\_element\_relationship objects. Each Data\_element\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.790 Date\_time\_interval\_assignment to Data\_element\_specification**

Each Date\_time\_interval\_assignment is applied to one or more Data\_element\_specification objects. Each Data\_element\_specification is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.791 Date\_time\_interval\_assignment to Design\_discipline\_item\_definition**

Each Date\_time\_interval\_assignment is applied to one or more Design\_discipline\_item\_definition objects. Each Design\_discipline\_item\_definition is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.792 Date\_time\_interval\_assignment to Device**

Each Date\_time\_interval\_assignment is applied to one or more Device objects. Each Device is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.793 Date\_time\_interval\_assignment to Device\_relationship**

Each Date\_time\_interval\_assignment is applied to one or more Device\_relationship objects. Each Device\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.794 Date\_time\_interval\_assignment to Document**

Each Date\_time\_interval\_assignment is applied to one or more Document objects. Each Document is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.795 Date\_time\_interval\_assignment to Document\_file**

Each Date\_time\_interval\_assignment is applied to one or more Document\_file objects. Each Document\_file is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.796 Date\_time\_interval\_assignment to Document\_file\_relationship**

Each Date\_time\_interval\_assignment is applied to one or more Document\_file\_relationship objects. Each Document\_file\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.797 Date\_time\_interval\_assignment to Document\_representation**

Each Date\_time\_interval\_assignment is applied to one or more Document\_representation objects. Each Document\_representation is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.798 Date\_time\_interval\_assignment to Document\_version**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Document\_version objects. Each Document\_version is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.799 Date\_time\_interval\_assignment to Document\_version - relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Document\_version\_relationship objects. Each Document\_version\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.800 Date\_time\_interval\_assignment to Drawing**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Drawing objects. Each Drawing is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.801 Date\_time\_interval\_assignment to Drawing\_sequence**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Drawing\_sequence objects. Each Drawing\_sequence is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.802 Date\_time\_interval\_assignment to Drawing\_sheet**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Drawing\_sheet objects. Each Drawing\_sheet is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.803 Date\_time\_interval\_assignment to Drawing\_sheet\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Drawing\_sheet\_relationship objects. Each Drawing\_sheet\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.804 Date\_time\_interval\_assignment to Free\_segment**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Free\_segment objects. Each Free\_segment is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.805 Date\_time\_interval\_assignment to Function\_definition**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Function\_definition objects. Each Function\_definition is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.806 Date\_time\_interval\_assignment to Function\_definition - relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Function\_definition\_relationship objects. Each Function\_definition\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.807 Date\_time\_interval\_assignment to Function\_interface**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Function\_interface objects. Each Function\_interface is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.808 Date\_time\_interval\_assignment to Function\_unit**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Function\_unit objects. Each Function\_unit is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.809 Date\_time\_interval\_assignment to Function\_unit\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Function\_unit\_relationship objects. Each Function\_unit\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.810 Date\_time\_interval\_assignment to Function\_version**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Function\_version objects. Each Function\_version is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.811 Date\_time\_interval\_assignment to Function\_version\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Function\_version\_relationship objects. Each Function\_version\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.812 Date\_time\_interval\_assignment to Functional\_connectivity\_definition**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Functional\_connectivity\_definition objects. Each Functional\_connectivity\_definition is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.813 Date\_time\_interval\_assignment to Functional\_connectivity\_definition\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Functional\_connectivity\_definition\_relationship objects. Each Functional\_connectivity\_definition\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.814 Date\_time\_interval\_assignment to Functional\_unit\_allocation**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Functional\_unit\_allocation objects. Each Functional\_unit\_allocation is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.815 Date\_time\_interval\_assignment to Functionality**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Functionality objects. Each Functionality is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.816 Date\_time\_interval\_assignment to General\_classification**

Each Date\_time\_interval\_assignment is\_applied\_to one or more General\_classification objects. Each General\_classification is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.817 Date\_time\_interval\_assignment to Generic\_note**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Generic\_note objects. Each Generic\_note is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.818 Date\_time\_interval\_assignment to Interface**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Interface objects. Each Interface is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.819 Date\_time\_interval\_assignment to Interface\_port**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Interface\_port objects. Each Interface\_port is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.820 Date\_time\_interval\_assignment to Interface\_terminal**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Interface\_terminal objects. Each Interface\_terminal is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.821 Date\_time\_interval\_assignment to Interval\_of\_time**

Each Date\_time\_interval\_assignment refers to exactly one Interval\_of\_time in the role of assigned\_time\_interval. Each Interval\_of\_time acts as assigned\_time\_interval for zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.822 Date\_time\_interval\_assignment to Item**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Item objects. Each Item is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.823 Date\_time\_interval\_assignment to Item\_definition\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Item\_definition\_relationship objects. Each Item\_definition\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.824 Date\_time\_interval\_assignment to Item\_version**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Item\_version objects. Each Item\_version is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.825 Date\_time\_interval\_assignment to Item\_version\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Item\_version\_relationship objects. Each Item\_version\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.826 Date\_time\_interval\_assignment to Location**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Location objects. Each Location is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.827 Date\_time\_interval\_assignment to Location\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Location\_relationship objects. Each Location\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.828 Date\_time\_interval\_assignment to Marking**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Marking objects. Each Marking is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.829 Date\_time\_interval\_assignment to Material**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Material objects. Each Material is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.830 Date\_time\_interval\_assignment to Node**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Node objects. Each Node is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.831 Date\_time\_interval\_assignment to Node\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Node\_relationship objects. Each Node\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.832 Date\_time\_interval\_assignment to Notification**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Notification objects. Each Notification is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.833 Date\_time\_interval\_assignment to Notification\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Notification\_relationship objects. Each Notification\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.834 Date\_time\_interval\_assignment to Offered\_function\_allocation**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Offered\_function\_allocation objects. Each Offered\_function\_allocation is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.835 Date\_time\_interval\_assignment to Organization\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Organization\_relationship objects. Each Organization\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.836 Date\_time\_interval\_assignment to Path**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Path objects. Each Path is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.837 Date\_time\_interval\_assignment to Path\_node**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Path\_node objects. Each Path\_node is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.838 Date\_time\_interval\_assignment to Path\_node\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Path\_node\_relationship objects. Each Path\_node\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.839 Date\_time\_interval\_assignment to Path\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Path\_relationship objects. Each Path\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.840 Date\_time\_interval\_assignment to Person\_in\_organization**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Person\_in\_organization objects. Each Person\_in\_organization is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.841 Date\_time\_interval\_assignment to Person\_in\_organization\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Person\_in\_organization\_relationship objects. Each Person\_in\_organization\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.842 Date\_time\_interval\_assignment to Physical\_assembly\_- relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Physical\_assembly\_relationship objects. Each Physical\_assembly\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.843 Date\_time\_interval\_assignment to Physical\_instance**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Physical\_instance objects. Each Physical\_instance is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.844 Date\_time\_interval\_assignment to Port**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Port objects. Each Port is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.845 Date\_time\_interval\_assignment to Port\_allocation**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Port\_allocation objects. Each Port\_allocation is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.846 Date\_time\_interval\_assignment to Port\_association**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Port\_association objects. Each Port\_association is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.847 Date\_time\_interval\_assignment to Preferred\_item\_allocation**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Preferred\_item\_allocation objects. Each Preferred\_item\_allocation is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.848 Date\_time\_interval\_assignment to Preferred\_item\_terminal\_- allocation**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Preferred\_item\_terminal\_allocation objects. Each Preferred\_item\_terminal\_allocation is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.849 Date\_time\_interval\_assignment to Process\_variable**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Process\_variable objects. Each Process\_variable is related to zero, one, or more Date\_time\_interval\_assignment objects.



**4.3.850 Date\_time\_interval\_assignment to Process\_variable\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Process\_variable\_relationship objects. Each Process\_variable\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.851 Date\_time\_interval\_assignment to Product\_class**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Product\_class objects. Each Product\_class is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.852 Date\_time\_interval\_assignment to Product\_identification**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Product\_identification objects. Each Product\_identification is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.853 Date\_time\_interval\_assignment to Product\_structure\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Product\_structure\_relationship objects. Each Product\_structure\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.854 Date\_time\_interval\_assignment to Project**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Project objects. Each Project is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.855 Date\_time\_interval\_assignment to Requirement**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Requirement objects. Each Requirement is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.856 Date\_time\_interval\_assignment to Requirement\_document\_assignment**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Requirement\_document\_assignment objects. Each Requirement\_document\_assignment is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.857 Date\_time\_interval\_assignment to Route**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Route objects. Each Route is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.858 Date\_time\_interval\_assignment to Route\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Route\_relationship objects. Each Route\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.859 Date\_time\_interval\_assignment to Routed\_segment**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Routed\_segment objects. Each Routed\_segment is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.860 Date\_time\_interval\_assignment to Section**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Section objects. Each Section is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.861 Date\_time\_interval\_assignment to Section\_end**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Section\_end objects. Each Section\_end is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.862 Date\_time\_interval\_assignment to Section\_interface**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Section\_interface objects. Each Section\_interface is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.863 Date\_time\_interval\_assignment to Section\_interface\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Section\_interface\_relationship objects. Each Section\_interface\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.864 Date\_time\_interval\_assignment to Section\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Section\_relationship objects. Each Section\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.865 Date\_time\_interval\_assignment to Security\_classification**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Security\_classification objects. Each Security\_classification is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.866 Date\_time\_interval\_assignment to Security\_level**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Security\_level objects. Each Security\_level is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.867 Date\_time\_interval\_assignment to Signal**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Signal objects. Each Signal is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.868 Date\_time\_interval\_assignment to Signal\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Signal\_relationship objects. Each Signal\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.869 Date\_time\_interval\_assignment to Signal\_value**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Signal\_value objects. Each Signal\_value is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.870 Date\_time\_interval\_assignment to Specification**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Specification objects. Each Specification is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.871 Date\_time\_interval\_assignment to Specification\_category**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Specification\_category objects. Each Specification\_category is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.872 Date\_time\_interval\_assignment to Specification\_expression**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Specification\_expression objects. Each Specification\_expression is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.873 Date\_time\_interval\_assignment to Specification\_inclusion**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Specification\_inclusion objects. Each Specification\_inclusion is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.874 Date\_time\_interval\_assignment to Technical\_system**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Technical\_system objects. Each Technical\_system is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.875 Date\_time\_interval\_assignment to Technical\_system\_relationship**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Technical\_system\_relationship objects. Each Technical\_system\_relationship is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.876 Date\_time\_interval\_assignment to Terminal**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Terminal objects. Each Terminal is related to zero, one, or more Date\_time\_interval\_assignment objects.

**4.3.877 Date\_time\_interval\_assignment to Work\_order**

Each Date\_time\_interval\_assignment is\_applied\_to one or more Work\_order objects. Each Work\_order is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.878 Date\_time\_interval\_assignment to Work\_request**

Each Date\_time\_interval\_assignment is applied to one or more Work\_request objects. Each Work\_request is related to zero, one, or more Date\_time\_interval\_assignment objects.

#### **4.3.879 Design\_discipline\_item\_definition to Application\_context**

Each Design\_discipline\_item\_definition refers to zero, one, or more Application\_context objects in the role of additional\_context. Each Application\_context acts as additional\_context for zero, one, or more Design\_discipline\_item\_definition objects.

#### **4.3.880 Design\_discipline\_item\_definition to Application\_context**

Each Design\_discipline\_item\_definition refers to exactly one Application\_context in the role of initial\_context. Each Application\_context acts as initial\_context for zero, one, or more Design\_discipline\_item\_definition objects.

#### **4.3.881 Design\_discipline\_item\_definition to Item\_version**

Each Design\_discipline\_item\_definition refers to exactly one Item\_version in the role of associated\_item\_version. Each Item\_version acts as associated\_item\_version for zero, one, or more Design\_discipline\_item\_definition objects.

#### **4.3.882 Detached\_representation\_reference to Annotation\_element**

Each Detached\_representation\_reference refers to exactly one Annotation\_element in the role of part\_of. Each Annotation\_element acts as part\_of for zero, one, or more Detached\_representation\_reference objects.

#### **4.3.883 Detached\_representation\_reference to Annotation\_element**

Each Detached\_representation\_reference refers to exactly one Annotation\_element in the role of refers\_to. Each Annotation\_element acts as refers\_to for zero, one, or more Detached\_representation\_reference objects.

#### **4.3.884 Device to Design\_discipline\_item\_definition**

Each Device refers to exactly one Design\_discipline\_item\_definition in the role of definition. Each Design\_discipline\_item\_definition acts as definition for zero, one, or more Device objects.

#### **4.3.885 Device to Object\_reference\_designation**

Each Device refers to zero or one Object\_reference\_designation in the role of extended\_designation. Each Object\_reference\_designation acts as extended\_designation for zero, one, or more Device objects.

#### **4.3.886 Device to Product\_identification**

Each Device refers to exactly one Product\_identification in the role of definition. Each Product\_identification acts as definition for zero, one, or more Device objects.

**4.3.887 Device\_relationship to Device**

Each Device\_relationship refers to exactly one Device in the role of related. Each Device acts as related for zero, one, or more Device\_relationship objects.

**4.3.888 Device\_relationship to Device**

Each Device\_relationship refers to exactly one Device in the role of relating. Each Device acts as relating for zero, one, or more Device\_relationship objects.

**4.3.889 Diameter\_dimension to Dimension\_line**

Each Diameter\_dimension refers to exactly one Dimension\_line in the role of extent. Each Dimension\_line acts as extent for zero, one, or more Diameter\_dimension objects.

**4.3.890 Diameter\_dimension to Projection\_line**

Each Diameter\_dimension refers to zero, one, or two Projection\_line objects in the role of component. Each Projection\_line acts as component for zero, one, or more Diameter\_dimension objects.

**4.3.891 Digital\_document to Digital\_file**

Each Digital\_document refers to zero, one, or more Digital\_file objects in the role of file. Each Digital\_file acts as file for zero, one, or more Digital\_document objects.

**4.3.892 Dimension\_callout to Dimension**

Each Dimension\_callout refers to zero or one Dimension in the role of defined\_primary\_dimension. Each Dimension acts as defined\_primary\_dimension for zero, one, or more Dimension\_callout objects.

**4.3.893 Dimension\_callout to Dimension**

Each Dimension\_callout refers to zero or one Dimension in the role of defined\_secondary\_dimension. Each Dimension acts as defined\_secondary\_dimension for zero, one, or more Dimension\_callout objects.

**4.3.894 Dimension\_line\_terminator to Annotation\_symbol**

Each Dimension\_line\_terminator refers to exactly one Annotation\_symbol in the role of symbol. Each Annotation\_symbol acts as symbol for zero, one, or more Dimension\_line\_terminator objects.

**4.3.895 Dimension\_line\_terminator to Dimension\_line**

Each Dimension\_line\_terminator refers to exactly one Dimension\_line in the role of line. Each Dimension\_line acts as line for zero, one, or more Dimension\_line\_terminator objects.

#### **4.3.896 Dimension\_sequence\_pair to Dimension**

Each Dimension\_sequence\_pair refers to exactly one Dimension in the role of predecessor. Each Dimension acts as predecessor for zero, one, or more Dimension\_sequence\_pair objects.

#### **4.3.897 Dimension\_sequence\_pair to Dimension**

Each Dimension\_sequence\_pair refers to exactly one Dimension in the role of successor. Each Dimension acts as successor for zero, one, or more Dimension\_sequence\_pair objects.

#### **4.3.898 Directed\_curve to Draughting\_callout**

Each Directed\_curve refers to zero or one Draughting\_callout in the role of directed\_callout. Each Draughting\_callout acts as directed\_callout for zero, one, or more Directed\_curve objects.

#### **4.3.899 Direction\_range to Connect\_area**

Each Direction\_range refers to exactly one Connect\_area in the role of associated\_connect\_area. Each Connect\_area acts as associated\_connect\_area for zero, one, or more Direction\_range objects.

#### **4.3.900 Document to Document\_designation**

Each Document refers to zero or one Document\_designation in the role of extended\_designation. Each Document\_designation acts as extended\_designation for zero, one, or more Document objects.

#### **4.3.901 Document\_assignment to Activity**

Each Document\_assignment is\_assigned\_to exactly one Activity. Each Activity is related to zero, one, or more Document\_assignment objects.

#### **4.3.902 Document\_assignment to Activity\_element**

Each Document\_assignment is\_assigned\_to exactly one Activity\_element. Each Activity\_element is related to zero, one, or more Document\_assignment objects.

#### **4.3.903 Document\_assignment to Activity\_method**

Each Document\_assignment is\_assigned\_to exactly one Activity\_method. Each Activity\_method is related to zero, one, or more Document\_assignment objects.

#### **4.3.904 Document\_assignment to Address**

Each Document\_assignment is\_assigned\_to exactly one Address. Each Address is related to zero, one, or more Document\_assignment objects.

#### **4.3.905 Document\_assignment to Approval**

Each Document\_assignment is\_assigned\_to exactly one Approval. Each Approval is related to zero, one, or more Document\_assignment objects.

**4.3.906 Document\_assignment to Approval\_status**

Each Document\_assignment is\_assigned\_to exactly one Approval\_status. Each Approval\_status is related to zero, one, or more Document\_assignment objects.

**4.3.907 Document\_assignment to Assembly\_component\_relationship**

Each Document\_assignment is\_assigned\_to exactly one Assembly\_component\_relationship. Each Assembly\_component\_relationship is related to zero, one, or more Document\_assignment objects.

**4.3.908 Document\_assignment to Cable\_pull\_information**

Each Document\_assignment is\_assigned\_to exactly one Cable\_pull\_information. Each Cable\_pull\_information is related to zero, one, or more Document\_assignment objects.

**4.3.909 Document\_assignment to Certification**

Each Document\_assignment is\_assigned\_to exactly one Certification. Each Certification is related to zero, one, or more Document\_assignment objects.

**4.3.910 Document\_assignment to Class\_category\_association**

Each Document\_assignment is\_assigned\_to exactly one Class\_category\_association. Each Class\_category\_association is related to zero, one, or more Document\_assignment objects.

**4.3.911 Document\_assignment to Class\_condition\_association**

Each Document\_assignment is\_assigned\_to exactly one Class\_condition\_association. Each Class\_condition\_association is related to zero, one, or more Document\_assignment objects.

**4.3.912 Document\_assignment to Class\_inclusion\_association**

Each Document\_assignment is\_assigned\_to exactly one Class\_inclusion\_association. Each Class\_inclusion\_association is related to zero, one, or more Document\_assignment objects.

**4.3.913 Document\_assignment to Class\_specification\_association**

Each Document\_assignment is\_assigned\_to exactly one Class\_specification\_association. Each Class\_specification\_association is related to zero, one, or more Document\_assignment objects.

**4.3.914 Document\_assignment to Classification\_association**

Each Document\_assignment is\_assigned\_to exactly one Classification\_association. Each Classification\_association is related to zero, one, or more Document\_assignment objects.

**4.3.915 Document\_assignment to Classification\_attribute**

Each Document\_assignment is\_assigned\_to exactly one Classification\_attribute. Each Classification\_attribute is related to zero, one, or more Document\_assignment objects.

#### **4.3.916 Document\_assignment to Classification\_system**

Each Document\_assignment is\_assigned\_to exactly one Classification\_system. Each Classification\_system is related to zero, one, or more Document\_assignment objects.

#### **4.3.917 Document\_assignment to Complex\_product**

Each Document\_assignment is\_assigned\_to exactly one Complex\_product. Each Complex\_product is related to zero, one, or more Document\_assignment objects.

#### **4.3.918 Document\_assignment to Connectivity\_definition**

Each Document\_assignment is\_assigned\_to exactly one Connectivity\_definition. Each Connectivity\_definition is related to zero, one, or more Document\_assignment objects.

#### **4.3.919 Document\_assignment to Contract**

Each Document\_assignment is\_assigned\_to exactly one Contract. Each Contract is related to zero, one, or more Document\_assignment objects.

#### **4.3.920 Document\_assignment to Data\_element**

Each Document\_assignment is\_assigned\_to exactly one Data\_element. Each Data\_element is related to zero, one, or more Document\_assignment objects.

#### **4.3.921 Document\_assignment to Data\_element\_definition**

Each Document\_assignment is\_assigned\_to exactly one Data\_element\_definition. Each Data\_element\_definition is related to zero, one, or more Document\_assignment objects.

#### **4.3.922 Document\_assignment to Data\_element\_specification**

Each Document\_assignment is\_assigned\_to exactly one Data\_element\_specification. Each Data\_element\_specification is related to zero, one, or more Document\_assignment objects.

#### **4.3.923 Document\_assignment to Descriptive\_specification**

Each Document\_assignment is\_assigned\_to exactly one Descriptive\_specification. Each Descriptive\_specification is related to zero, one, or more Document\_assignment objects.

#### **4.3.924 Document\_assignment to Design\_discipline\_item\_definition**

Each Document\_assignment is\_assigned\_to exactly one Design\_discipline\_item\_definition. Each Design\_discipline\_item\_definition is related to zero, one, or more Document\_assignment objects.

#### **4.3.925 Document\_assignment to Device**

Each Document\_assignment is\_assigned\_to exactly one Device. Each Device is related to zero, one, or more Document\_assignment objects.



#### **4.3.926 Document\_assignment to Document**

Each Document\_assignment refers to exactly one Document in the role of assigned\_document. Each Document acts as assigned\_document for zero, one, or more Document\_assignment objects.

#### **4.3.927 Document\_assignment to Document\_file**

Each Document\_assignment refers to exactly one Document\_file in the role of assigned\_document. Each Document\_file acts as assigned\_document for zero, one, or more Document\_assignment objects.

#### **4.3.928 Document\_assignment to Document\_representation**

Each Document\_assignment refers to exactly one Document\_representation in the role of assigned\_document. Each Document\_representation acts as assigned\_document for zero, one, or more Document\_assignment objects.

#### **4.3.929 Document\_assignment to Document\_version**

Each Document\_assignment refers to exactly one Document\_version in the role of assigned\_document. Each Document\_version acts as assigned\_document for zero, one, or more Document\_assignment objects.

#### **4.3.930 Document\_assignment to Drawing**

Each Document\_assignment is\_assigned\_to exactly one Drawing. Each Drawing is related to zero, one, or more Document\_assignment objects.

#### **4.3.931 Document\_assignment to Drawing\_sheet**

Each Document\_assignment is\_assigned\_to exactly one Drawing\_sheet. Each Drawing\_sheet is related to zero, one, or more Document\_assignment objects.

#### **4.3.932 Document\_assignment to Drawing\_view**

Each Document\_assignment is\_assigned\_to exactly one Drawing\_view. Each Drawing\_view is related to zero, one, or more Document\_assignment objects.

#### **4.3.933 Document\_assignment to Function\_definition**

Each Document\_assignment is\_assigned\_to exactly one Function\_definition. Each Function\_definition is related to zero, one, or more Document\_assignment objects.

#### **4.3.934 Document\_assignment to Function\_interface**

Each Document\_assignment is\_assigned\_to exactly one Function\_interface. Each Function\_interface is related to zero, one, or more Document\_assignment objects.

#### **4.3.935 Document\_assignment to Function\_unit**

Each Document\_assignment is\_assigned\_to exactly one Function\_unit. Each Function\_unit is related to zero, one, or more Document\_assignment objects.

#### **4.3.936 Document\_assignment to Function\_version**

Each Document\_assignment is\_assigned\_to exactly one Function\_version. Each Function\_version is related to zero, one, or more Document\_assignment objects.

#### **4.3.937 Document\_assignment to Functional\_connectivity\_definition**

Each Document\_assignment is\_assigned\_to exactly one Functional\_connectivity\_definition. Each Functional\_connectivity\_definition is related to zero, one, or more Document\_assignment objects.

#### **4.3.938 Document\_assignment to Functionality**

Each Document\_assignment is\_assigned\_to exactly one Functionality. Each Functionality is related to zero, one, or more Document\_assignment objects.

#### **4.3.939 Document\_assignment to General\_classification**

Each Document\_assignment is\_assigned\_to exactly one General\_classification. Each General\_classification is related to zero, one, or more Document\_assignment objects.

#### **4.3.940 Document\_assignment to General\_classification**

Each Document\_assignment is\_assigned\_to exactly one General\_classification. Each General\_classification is related to zero, one, or more Document\_assignment objects.

#### **4.3.941 Document\_assignment to Generic\_note**

Each Document\_assignment is\_assigned\_to exactly one Generic\_note. Each Generic\_note is related to zero, one, or more Document\_assignment objects.

#### **4.3.942 Document\_assignment to Interface**

Each Document\_assignment is\_assigned\_to exactly one Interface. Each Interface is related to zero, one, or more Document\_assignment objects.

#### **4.3.943 Document\_assignment to Interface\_port**

Each Document\_assignment is\_assigned\_to exactly one Interface\_port. Each Interface\_port is related to zero, one, or more Document\_assignment objects.

#### **4.3.944 Document\_assignment to Interface\_terminal**

Each Document\_assignment is\_assigned\_to exactly one Interface\_terminal. Each Interface\_terminal is related to zero, one, or more Document\_assignment objects.

**4.3.945 Document\_assignment to Item**

Each Document\_assignment is\_assigned\_to exactly one Item. Each Item is related to zero, one, or more Document\_assignment objects.

**4.3.946 Document\_assignment to Item\_definition\_relationship**

Each Document\_assignment is\_assigned\_to exactly one Item\_definition\_relationship. Each Item\_definition\_relationship is related to zero, one, or more Document\_assignment objects.

**4.3.947 Document\_assignment to Item\_identification**

Each Document\_assignment is\_assigned\_to exactly one Item\_identification. Each Item\_identification is related to zero, one, or more Document\_assignment objects.

**4.3.948 Document\_assignment to Item\_version**

Each Document\_assignment is\_assigned\_to exactly one Item\_version. Each Item\_version is related to zero, one, or more Document\_assignment objects.

**4.3.949 Document\_assignment to Location**

Each Document\_assignment is\_assigned\_to exactly one Location. Each Location is related to zero, one, or more Document\_assignment objects.

**4.3.950 Document\_assignment to Marking**

Each Document\_assignment is\_assigned\_to exactly one Marking. Each Marking is related to zero, one, or more Document\_assignment objects.

**4.3.951 Document\_assignment to Node**

Each Document\_assignment is\_assigned\_to exactly one Node. Each Node is related to zero, one, or more Document\_assignment objects.

**4.3.952 Document\_assignment to Notification**

Each Document\_assignment is\_assigned\_to exactly one Notification. Each Notification is related to zero, one, or more Document\_assignment objects.

**4.3.953 Document\_assignment to Object\_designation**

Each Document\_assignment is\_assigned\_to exactly one Object\_designation. Each Object\_designation is related to zero, one, or more Document\_assignment objects.

**4.3.954 Document\_assignment to Organization**

Each Document\_assignment is\_assigned\_to exactly one Organization. Each Organization is related to zero, one, or more Document\_assignment objects.

#### **4.3.955 Document\_assignment to Path**

Each Document\_assignment is\_assigned\_to exactly one Path. Each Path is related to zero, one, or more Document\_assignment objects.

#### **4.3.956 Document\_assignment to Path\_node**

Each Document\_assignment is\_assigned\_to exactly one Path\_node. Each Path\_node is related to zero, one, or more Document\_assignment objects.

#### **4.3.957 Document\_assignment to Person**

Each Document\_assignment is\_assigned\_to exactly one Person. Each Person is related to zero, one, or more Document\_assignment objects.

#### **4.3.958 Document\_assignment to Physical\_assembly\_relationship**

Each Document\_assignment is\_assigned\_to exactly one Physical\_assembly\_relationship. Each Physical\_assembly\_relationship is related to zero, one, or more Document\_assignment objects.

#### **4.3.959 Document\_assignment to Physical\_instance**

Each Document\_assignment is\_assigned\_to exactly one Physical\_instance. Each Physical\_instance is related to zero, one, or more Document\_assignment objects.

#### **4.3.960 Document\_assignment to Port**

Each Document\_assignment is\_assigned\_to exactly one Port. Each Port is related to zero, one, or more Document\_assignment objects.

#### **4.3.961 Document\_assignment to Process\_variable**

Each Document\_assignment is\_assigned\_to exactly one Process\_variable. Each Process\_variable is related to zero, one, or more Document\_assignment objects.

#### **4.3.962 Document\_assignment to Product\_class**

Each Document\_assignment is\_assigned\_to exactly one Product\_class. Each Product\_class is related to zero, one, or more Document\_assignment objects.

#### **4.3.963 Document\_assignment to Product\_identification**

Each Document\_assignment is\_assigned\_to exactly one Product\_identification. Each Product\_identification is related to zero, one, or more Document\_assignment objects.

#### **4.3.964 Document\_assignment to Product\_structure\_relationship**

Each Document\_assignment is\_assigned\_to exactly one Product\_structure\_relationship. Each Product\_structure\_relationship is related to zero, one, or more Document\_assignment objects.

#### **4.3.965 Document\_assignment to Project**

Each Document\_assignment is\_assigned\_to exactly one Project. Each Project is related to zero, one, or more Document\_assignment objects.

#### **4.3.966 Document\_assignment to Retention\_period**

Each Document\_assignment is\_assigned\_to exactly one Retention\_period. Each Retention\_period is related to zero, one, or more Document\_assignment objects.

#### **4.3.967 Document\_assignment to Route**

Each Document\_assignment is\_assigned\_to exactly one Route. Each Route is related to zero, one, or more Document\_assignment objects.

#### **4.3.968 Document\_assignment to Section**

Each Document\_assignment is\_assigned\_to exactly one Section. Each Section is related to zero, one, or more Document\_assignment objects.

#### **4.3.969 Document\_assignment to Section\_end**

Each Document\_assignment is\_assigned\_to exactly one Section\_end. Each Section\_end is related to zero, one, or more Document\_assignment objects.

#### **4.3.970 Document\_assignment to Section\_interface**

Each Document\_assignment is\_assigned\_to exactly one Section\_interface. Each Section\_interface is related to zero, one, or more Document\_assignment objects.

#### **4.3.971 Document\_assignment to Security\_classification**

Each Document\_assignment is\_assigned\_to exactly one Security\_classification. Each Security\_classification is related to zero, one, or more Document\_assignment objects.

#### **4.3.972 Document\_assignment to Security\_level**

Each Document\_assignment is\_assigned\_to exactly one Security\_level. Each Security\_level is related to zero, one, or more Document\_assignment objects.

#### **4.3.973 Document\_assignment to Signal**

Each Document\_assignment is\_assigned\_to exactly one Signal. Each Signal is related to zero, one, or more Document\_assignment objects.

#### **4.3.974 Document\_assignment to Signal\_value**

Each Document\_assignment is\_assigned\_to exactly one Signal\_value. Each Signal\_value is related to zero, one, or more Document\_assignment objects.

#### **4.3.975 Document\_assignment to Specification**

Each Document\_assignment is\_assigned\_to exactly one Specification. Each Specification is related to zero, one, or more Document\_assignment objects.

#### **4.3.976 Document\_assignment to Specification\_category**

Each Document\_assignment is\_assigned\_to exactly one Specification\_category. Each Specification\_category is related to zero, one, or more Document\_assignment objects.

#### **4.3.977 Document\_assignment to Specification\_expression**

Each Document\_assignment is\_assigned\_to exactly one Specification\_expression. Each Specification\_expression is related to zero, one, or more Document\_assignment objects.

#### **4.3.978 Document\_assignment to Specification\_inclusion**

Each Document\_assignment is\_assigned\_to exactly one Specification\_inclusion. Each Specification\_inclusion is related to zero, one, or more Document\_assignment objects.

#### **4.3.979 Document\_assignment to Technical\_system**

Each Document\_assignment is\_assigned\_to exactly one Technical\_system. Each Technical\_system is related to zero, one, or more Document\_assignment objects.

#### **4.3.980 Document\_assignment to Terminal**

Each Document\_assignment is\_assigned\_to exactly one Terminal. Each Terminal is related to zero, one, or more Document\_assignment objects.

#### **4.3.981 Document\_assignment to Work\_order**

Each Document\_assignment is\_assigned\_to exactly one Work\_order. Each Work\_order is related to zero, one, or more Document\_assignment objects.

#### **4.3.982 Document\_assignment to Work\_request**

Each Document\_assignment is\_assigned\_to exactly one Work\_request. Each Work\_request is related to zero, one, or more Document\_assignment objects.

#### **4.3.983 Document\_content\_property to Language**

Each Document\_content\_property refers to zero, one, or more Language objects in the role of languages. Each Language acts as languages for zero, one, or more Document\_content\_property objects.

**4.3.984 Document\_content\_property to Numerical\_value**

Each Document\_content\_property refers to zero or one Numerical\_value in the role of real\_world\_scale. Each Numerical\_value acts as real\_world\_scale for zero, one, or more Document\_content\_property objects.

**4.3.985 Document\_file to Document\_content\_property**

Each Document\_file refers to zero or one Document\_content\_property in the role of content. Each Document\_content\_property acts as content for zero, one, or more Document\_file objects.

**4.3.986 Document\_file to Document\_creation\_property**

Each Document\_file refers to zero or one Document\_creation\_property in the role of creation. Each Document\_creation\_property acts as creation for zero, one, or more Document\_file objects.

**4.3.987 Document\_file to Document\_format\_property**

Each Document\_file refers to zero or one Document\_format\_property in the role of file\_format. Each Document\_format\_property acts as file\_format for zero, one, or more Document\_file objects.

**4.3.988 Document\_file to Document\_size\_property**

Each Document\_file refers to zero or one Document\_size\_property in the role of size. Each Document\_size\_property acts as size for zero, one, or more Document\_file objects.

**4.3.989 Document\_file to Document\_type\_property**

Each Document\_file refers to zero or one Document\_type\_property in the role of document\_file\_type. Each Document\_type\_property acts as document\_file\_type for zero, one, or more Document\_file objects.

**4.3.990 Document\_file to External\_file\_id\_and\_location**

Each Document\_file refers to zero, one, or more External\_file\_id\_and\_location objects in the role of external\_id\_and\_location. Each External\_file\_id\_and\_location acts as external\_id\_and\_location for zero, one, or more Document\_file objects.

**4.3.991 Document\_file\_relationship to Document\_file**

Each Document\_file\_relationship refers to exactly one Document\_file in the role of related. Each Document\_file acts as related for zero, one, or more Document\_file\_relationship objects.

**4.3.992 Document\_file\_relationship to Document\_file**

Each Document\_file\_relationship refers to exactly one Document\_file in the role of relating. Each Document\_file acts as relating for zero, one, or more Document\_file\_relationship objects.

#### **4.3.993 Document\_format\_property to Rectangular\_size**

Each Document\_format\_property refers to exactly one Rectangular\_size in the role of size\_format. Each Rectangular\_size acts as size\_format for zero, one, or more Document\_format\_property objects.

#### **4.3.994 Document\_representation to Document\_content\_property**

Each Document\_representation refers to zero or one Document\_content\_property in the role of content. Each Document\_content\_property acts as content for zero, one, or more Document\_representation objects.

#### **4.3.995 Document\_representation to Document\_creation\_property**

Each Document\_representation refers to zero or one Document\_creation\_property in the role of creation. Each Document\_creation\_property acts as creation for zero, one, or more Document\_representation objects.

#### **4.3.996 Document\_representation to Document\_format\_property**

Each Document\_representation refers to zero or one Document\_format\_property in the role of representation\_format. Each Document\_format\_property acts as representation\_format for zero, one, or more Document\_representation objects.

#### **4.3.997 Document\_representation to Document\_location\_property**

Each Document\_representation refers to zero, one, or more Document\_location\_property objects in the role of common\_location. Each Document\_location\_property acts as common\_location for zero, one, or more Document\_representation objects.

#### **4.3.998 Document\_representation to Document\_size\_property**

Each Document\_representation refers to zero or one Document\_size\_property in the role of size. Each Document\_size\_property acts as size for zero, one, or more Document\_representation objects.

#### **4.3.999 Document\_representation to Document\_version**

Each Document\_representation refers to exactly one Document\_version in the role of associated\_document\_version. Each Document\_version acts as associated\_document\_version for zero, one, or more Document\_representation objects.

#### **4.3.1000 Document\_size\_property to Numerical\_value**

Each Document\_size\_property refers to zero or one Numerical\_value in the role of file\_size. Each Numerical\_value acts as file\_size for zero, one, or more Document\_size\_property objects.

#### **4.3.1001 Document\_size\_property to Numerical\_value**

Each Document\_size\_property refers to zero or one Numerical\_value in the role of page\_count. Each Numerical\_value acts as page\_count for zero, one, or more Document\_size\_property objects.



**4.3.1002 Document\_structure to Document\_representation**

Each Document\_structure refers to exactly one Document\_representation in the role of related. Each Document\_representation acts as related for zero, one, or more Document\_structure objects.

**4.3.1003 Document\_structure to Document\_representation**

Each Document\_structure refers to exactly one Document\_representation in the role of relating. Each Document\_representation acts as relating for zero, one, or more Document\_structure objects.

**4.3.1004 Document\_type\_property to Classification\_system**

Each Document\_type\_property refers to zero or one Classification\_system in the role of used\_classification\_system. Each Classification\_system acts as used\_classification\_system for zero, one, or more Document\_type\_property objects.

**4.3.1005 Document\_version to Document**

Each Document\_version refers to exactly one Document in the role of associated\_document. Each Document acts as associated\_document for zero, one, or more Document\_version objects.

**4.3.1006 Document\_version\_relationship to Document\_version**

Each Document\_version\_relationship refers to exactly one Document\_version in the role of related. Each Document\_version acts as related for zero, one, or more Document\_version\_relationship objects.

**4.3.1007 Document\_version\_relationship to Document\_version**

Each Document\_version\_relationship refers to exactly one Document\_version in the role of relating. Each Document\_version acts as relating for zero, one, or more Document\_version\_relationship objects.

**4.3.1008 Draughting\_callout to Annotation\_curve**

Each Draughting\_callout refers to one or more Annotation\_curve objects in the role of components. Each Annotation\_curve acts as components for zero, one, or more Draughting\_callout objects.

**4.3.1009 Draughting\_callout to Annotation\_symbol**

Each Draughting\_callout refers to one or more Annotation\_symbol objects in the role of components. Each Annotation\_symbol acts as components for zero, one, or more Draughting\_callout objects.

**4.3.1010 Draughting\_callout to Text**

Each Draughting\_callout refers to one or more Text objects in the role of components. Each Text acts as components for zero, one, or more Draughting\_callout objects.

#### **4.3.1011 Draughting\_model to Cartesian\_coordinate\_space\_2d**

Each Draughting\_model refers to exactly one Cartesian\_coordinate\_space\_2d in the role of coordinate\_space. Each Cartesian\_coordinate\_space\_2d acts as coordinate\_space for zero, one, or more Draughting\_model objects.

#### **4.3.1012 Draughting\_model to Model\_placed\_annotation**

Each Draughting\_model refers to one or more Model\_placed\_annotation objects in the role of element. Each Model\_placed\_annotation acts as element for zero, one, or more Draughting\_model objects.

#### **4.3.1013 Drawing to Document\_designation**

Each Drawing refers to zero or one Document\_designation in the role of extended\_designation. Each Document\_designation acts as extended\_designation for zero, one, or more Drawing objects.

#### **4.3.1014 Drawing\_assignment to Activity**

Each Drawing\_assignment is\_assigned\_to exactly one Activity. Each Activity is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1015 Drawing\_assignment to Address**

Each Drawing\_assignment is\_assigned\_to exactly one Address. Each Address is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1016 Drawing\_assignment to Approval**

Each Drawing\_assignment is\_assigned\_to exactly one Approval. Each Approval is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1017 Drawing\_assignment to Approval\_status**

Each Drawing\_assignment is\_assigned\_to exactly one Approval\_status. Each Approval\_status is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1018 Drawing\_assignment to Cable\_pull\_information**

Each Drawing\_assignment is\_assigned\_to exactly one Cable\_pull\_information. Each Cable\_pull\_information is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1019 Drawing\_assignment to Class\_category\_association**

Each Drawing\_assignment is\_assigned\_to exactly one Class\_category\_association. Each Class\_category\_association is related to zero, one, or more Drawing\_assignment objects.

**4.3.1020 Drawing\_assignment to Class\_condition\_association**

Each Drawing\_assignment is\_assigned\_to exactly one Class\_condition\_association. Each Class\_condition\_association is related to zero, one, or more Drawing\_assignment objects.

**4.3.1021 Drawing\_assignment to Class\_inclusion\_association**

Each Drawing\_assignment is\_assigned\_to exactly one Class\_inclusion\_association. Each Class\_inclusion\_association is related to zero, one, or more Drawing\_assignment objects.

**4.3.1022 Drawing\_assignment to Class\_specification\_association**

Each Drawing\_assignment is\_assigned\_to exactly one Class\_specification\_association. Each Class\_specification\_association is related to zero, one, or more Drawing\_assignment objects.

**4.3.1023 Drawing\_assignment to Classification\_attribute**

Each Drawing\_assignment is\_assigned\_to exactly one Classification\_attribute. Each Classification\_attribute is related to zero, one, or more Drawing\_assignment objects.

**4.3.1024 Drawing\_assignment to Classification\_system**

Each Drawing\_assignment is\_assigned\_to exactly one Classification\_system. Each Classification\_system is related to zero, one, or more Drawing\_assignment objects.

**4.3.1025 Drawing\_assignment to Complex\_product**

Each Drawing\_assignment is\_assigned\_to exactly one Complex\_product. Each Complex\_product is related to zero, one, or more Drawing\_assignment objects.

**4.3.1026 Drawing\_assignment to Connectivity\_definition**

Each Drawing\_assignment is\_assigned\_to exactly one Connectivity\_definition. Each Connectivity\_definition is related to zero, one, or more Drawing\_assignment objects.

**4.3.1027 Drawing\_assignment to Contract**

Each Drawing\_assignment is\_assigned\_to exactly one Contract. Each Contract is related to zero, one, or more Drawing\_assignment objects.

**4.3.1028 Drawing\_assignment to Data\_element**

Each Drawing\_assignment is\_assigned\_to exactly one Data\_element. Each Data\_element is related to zero, one, or more Drawing\_assignment objects.

**4.3.1029 Drawing\_assignment to Data\_element\_definition**

Each Drawing\_assignment is\_assigned\_to exactly one Data\_element\_definition. Each Data\_element\_definition is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1030 Drawing\_assignment to Data\_element\_specification**

Each Drawing\_assignment is\_assigned\_to exactly one Data\_element\_specification. Each Data\_element\_specification is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1031 Drawing\_assignment to Design\_discipline\_item\_definition**

Each Drawing\_assignment is\_assigned\_to exactly one Design\_discipline\_item\_definition. Each Design\_discipline\_item\_definition is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1032 Drawing\_assignment to Device**

Each Drawing\_assignment is\_assigned\_to exactly one Device. Each Device is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1033 Drawing\_assignment to Drawing**

Each Drawing\_assignment refers to exactly one Drawing in the role of assigned\_drawing. Each Drawing acts as assigned\_drawing for zero, one, or more Drawing\_assignment objects.

#### **4.3.1034 Drawing\_assignment to Function\_definition**

Each Drawing\_assignment is\_assigned\_to exactly one Function\_definition. Each Function\_definition is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1035 Drawing\_assignment to Function\_interface**

Each Drawing\_assignment is\_assigned\_to exactly one Function\_interface. Each Function\_interface is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1036 Drawing\_assignment to Function\_unit**

Each Drawing\_assignment is\_assigned\_to exactly one Function\_unit. Each Function\_unit is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1037 Drawing\_assignment to Function\_version**

Each Drawing\_assignment is\_assigned\_to exactly one Function\_version. Each Function\_version is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1038 Drawing\_assignment to Functional\_connectivity\_definition**

Each Drawing\_assignment is\_assigned\_to exactly one Functional\_connectivity\_definition. Each Functional\_connectivity\_definition is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1039 Drawing\_assignment to Functionality**

Each Drawing\_assignment is\_assigned\_to exactly one Functionality. Each Functionality is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1040 Drawing\_assignment to General\_classification**

Each Drawing\_assignment is\_assigned\_to exactly one General\_classification. Each General\_classification is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1041 Drawing\_assignment to Generic\_note**

Each Drawing\_assignment is\_assigned\_to exactly one Generic\_note. Each Generic\_note is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1042 Drawing\_assignment to Interface**

Each Drawing\_assignment is\_assigned\_to exactly one Interface. Each Interface is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1043 Drawing\_assignment to Interface\_port**

Each Drawing\_assignment is\_assigned\_to exactly one Interface\_port. Each Interface\_port is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1044 Drawing\_assignment to Interface\_terminal**

Each Drawing\_assignment is\_assigned\_to exactly one Interface\_terminal. Each Interface\_terminal is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1045 Drawing\_assignment to Item**

Each Drawing\_assignment is\_assigned\_to exactly one Item. Each Item is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1046 Drawing\_assignment to Item\_identification**

Each Drawing\_assignment is\_assigned\_to exactly one Item\_identification. Each Item\_identification is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1047 Drawing\_assignment to Item\_version**

Each Drawing\_assignment is\_assigned\_to exactly one Item\_version. Each Item\_version is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1048 Drawing\_assignment to Location**

Each Drawing\_assignment is\_assigned\_to exactly one Location. Each Location is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1049 Drawing\_assignment to Marking**

Each Drawing\_assignment is\_assigned\_to exactly one Marking. Each Marking is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1050 Drawing\_assignment to Node**

Each Drawing\_assignment is\_assigned\_to exactly one Node. Each Node is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1051 Drawing\_assignment to Notification**

Each Drawing\_assignment is\_assigned\_to exactly one Notification. Each Notification is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1052 Drawing\_assignment to Object\_designation**

Each Drawing\_assignment is\_assigned\_to exactly one Object\_designation. Each Object\_designation is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1053 Drawing\_assignment to Organization**

Each Drawing\_assignment is\_assigned\_to exactly one Organization. Each Organization is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1054 Drawing\_assignment to Path**

Each Drawing\_assignment is\_assigned\_to exactly one Path. Each Path is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1055 Drawing\_assignment to Path\_node**

Each Drawing\_assignment is\_assigned\_to exactly one Path\_node. Each Path\_node is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1056 Drawing\_assignment to Person**

Each Drawing\_assignment is\_assigned\_to exactly one Person. Each Person is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1057 Drawing\_assignment to Physical\_assembly\_relationship**

Each Drawing\_assignment is\_assigned\_to exactly one Physical\_assembly\_relationship. Each Physical\_assembly\_relationship is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1058 Drawing\_assignment to Physical\_instance**

Each Drawing\_assignment is\_assigned\_to exactly one Physical\_instance. Each Physical\_instance is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1059 Drawing\_assignment to Port**

Each Drawing\_assignment is\_assigned\_to exactly one Port. Each Port is related to zero, one, or more Drawing\_assignment objects.

**4.3.1060 Drawing\_assignment to Process\_variable**

Each Drawing\_assignment is\_assigned\_to exactly one Process\_variable. Each Process\_variable is related to zero, one, or more Drawing\_assignment objects.

**4.3.1061 Drawing\_assignment to Product\_class**

Each Drawing\_assignment is\_assigned\_to exactly one Product\_class. Each Product\_class is related to zero, one, or more Drawing\_assignment objects.

**4.3.1062 Drawing\_assignment to Product\_identification**

Each Drawing\_assignment is\_assigned\_to exactly one Product\_identification. Each Product\_identification is related to zero, one, or more Drawing\_assignment objects.

**4.3.1063 Drawing\_assignment to Project**

Each Drawing\_assignment is\_assigned\_to exactly one Project. Each Project is related to zero, one, or more Drawing\_assignment objects.

**4.3.1064 Drawing\_assignment to Retention\_period**

Each Drawing\_assignment is\_assigned\_to exactly one Retention\_period. Each Retention\_period is related to zero, one, or more Drawing\_assignment objects.

**4.3.1065 Drawing\_assignment to Route**

Each Drawing\_assignment is\_assigned\_to exactly one Route. Each Route is related to zero, one, or more Drawing\_assignment objects.

**4.3.1066 Drawing\_assignment to Section**

Each Drawing\_assignment is\_assigned\_to exactly one Section. Each Section is related to zero, one, or more Drawing\_assignment objects.

**4.3.1067 Drawing\_assignment to Section\_end**

Each Drawing\_assignment is\_assigned\_to exactly one Section\_end. Each Section\_end is related to zero, one, or more Drawing\_assignment objects.

**4.3.1068 Drawing\_assignment to Section\_interface**

Each Drawing\_assignment is\_assigned\_to exactly one Section\_interface. Each Section\_interface is related to zero, one, or more Drawing\_assignment objects.

**4.3.1069 Drawing\_assignment to Security\_classification**

Each Drawing\_assignment is\_assigned\_to exactly one Security\_classification. Each Security\_classification is related to zero, one, or more Drawing\_assignment objects.



#### **4.3.1070 Drawing\_assignment to Security\_level**

Each Drawing\_assignment is\_assigned\_to exactly one Security\_level. Each Security\_level is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1071 Drawing\_assignment to Signal**

Each Drawing\_assignment is\_assigned\_to exactly one Signal. Each Signal is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1072 Drawing\_assignment to Signal\_value**

Each Drawing\_assignment is\_assigned\_to exactly one Signal\_value. Each Signal\_value is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1073 Drawing\_assignment to Specification**

Each Drawing\_assignment is\_assigned\_to exactly one Specification. Each Specification is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1074 Drawing\_assignment to Specification\_category**

Each Drawing\_assignment is\_assigned\_to exactly one Specification\_category. Each Specification\_category is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1075 Drawing\_assignment to Specification\_expression**

Each Drawing\_assignment is\_assigned\_to exactly one Specification\_expression. Each Specification\_expression is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1076 Drawing\_assignment to Specification\_inclusion**

Each Drawing\_assignment is\_assigned\_to exactly one Specification\_inclusion. Each Specification\_inclusion is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1077 Drawing\_assignment to Technical\_system**

Each Drawing\_assignment is\_assigned\_to exactly one Technical\_system. Each Technical\_system is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1078 Drawing\_assignment to Terminal**

Each Drawing\_assignment is\_assigned\_to exactly one Terminal. Each Terminal is related to zero, one, or more Drawing\_assignment objects.

#### **4.3.1079 Drawing\_assignment to Work\_order**

Each Drawing\_assignment is\_assigned\_to exactly one Work\_order. Each Work\_order is related to zero, one, or more Drawing\_assignment objects.



**4.3.1080 Drawing\_assignment to Work\_request**

Each Drawing\_assignment is\_assigned\_to exactly one Work\_request. Each Work\_request is related to zero, one, or more Drawing\_assignment objects.

**4.3.1081 Drawing\_sequence to Drawing**

Each Drawing\_sequence refers to exactly one Drawing in the role of following\_version. Each Drawing acts as following\_version for zero, one, or more Drawing\_sequence objects.

**4.3.1082 Drawing\_sequence to Drawing**

Each Drawing\_sequence refers to exactly one Drawing in the role of preceding\_version. Each Drawing acts as preceding\_version for zero, one, or more Drawing\_sequence objects.

**4.3.1083 Drawing\_sheet to Cartesian\_coordinate\_space\_2d**

Each Drawing\_sheet refers to exactly one Cartesian\_coordinate\_space\_2d in the role of coordinate\_space. Each Cartesian\_coordinate\_space\_2d acts as coordinate\_space for zero, one, or more Drawing\_sheet objects.

**4.3.1084 Drawing\_sheet to Document\_designation**

Each Drawing\_sheet refers to zero or one Document\_designation in the role of extended\_designation. Each Document\_designation acts as extended\_designation for zero, one, or more Drawing\_sheet objects.

**4.3.1085 Drawing\_sheet to Drawing**

Each Drawing\_sheet refers to exactly one Drawing in the role of associated\_drawing. Each Drawing acts as associated\_drawing for zero, one, or more Drawing\_sheet objects.

**4.3.1086 Drawing\_sheet to Rectangular\_area**

Each Drawing\_sheet refers to exactly one Rectangular\_area in the role of size. Each Rectangular\_area acts as size for zero, one, or more Drawing\_sheet objects.

**4.3.1087 Drawing\_sheet\_relationship to Drawing\_sheet**

Each Drawing\_sheet\_relationship refers to exactly one Drawing\_sheet in the role of related. Each Drawing\_sheet acts as related for zero, one, or more Drawing\_sheet\_relationship objects.

**4.3.1088 Drawing\_sheet\_relationship to Drawing\_sheet**

Each Drawing\_sheet\_relationship refers to exactly one Drawing\_sheet in the role of relating. Each Drawing\_sheet acts as relating for zero, one, or more Drawing\_sheet\_relationship objects.

#### **4.3.1089 Drawing\_view to Cartesian\_coordinate\_space\_2d**

Each Drawing\_view refers to exactly one Cartesian\_coordinate\_space\_2d in the role of coordinate\_space. Each Cartesian\_coordinate\_space\_2d acts as coordinate\_space for zero, one, or more Drawing\_view objects.

#### **4.3.1090 Drawing\_view to Drawing\_sheet**

Each Drawing\_view refers to exactly one Drawing\_sheet in the role of containing\_sheet. Each Drawing\_sheet acts as containing\_sheet for zero, one, or more Drawing\_view objects.

#### **4.3.1091 Drawing\_view to Point\_2d**

Each Drawing\_view refers to exactly one Point\_2d in the role of position. Each Point\_2d acts as position for zero, one, or more Drawing\_view objects.

#### **4.3.1092 Effectivity to Date\_time**

Each Effectivity refers to zero or one Date\_time in the role of end\_definition. Each Date\_time acts as end\_definition for zero, one, or more Effectivity objects.

#### **4.3.1093 Effectivity to Date\_time**

Each Effectivity refers to zero or one Date\_time in the role of start\_definition. Each Date\_time acts as start\_definition for zero, one, or more Effectivity objects.

#### **4.3.1094 Effectivity to Duration**

Each Effectivity refers to zero or one Duration in the role of period. Each Duration acts as period for zero, one, or more Effectivity objects.

#### **4.3.1095 Effectivity to Event\_reference**

Each Effectivity refers to zero or one Event\_reference in the role of end\_definition. Each Event\_reference acts as end\_definition for zero, one, or more Effectivity objects.

#### **4.3.1096 Effectivity to Event\_reference**

Each Effectivity refers to zero or one Event\_reference in the role of start\_definition. Each Event\_reference acts as start\_definition for zero, one, or more Effectivity objects.

#### **4.3.1097 Effectivity to Organization**

Each Effectivity refers to zero, one, or more Organization objects in the role of concerned\_organization. Each Organization acts as concerned\_organization for zero, one, or more Effectivity objects.

**4.3.1098 Effectivity\_assignment to Activity\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Activity\_relationship. Each Activity\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1099 Effectivity\_assignment to Alternate\_item\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Alternate\_item\_relationship. Each Alternate\_item\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1100 Effectivity\_assignment to Assembly\_component\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Assembly\_component\_relationship. Each Assembly\_component\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1101 Effectivity\_assignment to Assembly\_substitute\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Assembly\_substitute\_relationship. Each Assembly\_substitute\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1102 Effectivity\_assignment to Cable\_pull\_information**

Each Effectivity\_assignment is\_applied\_to exactly one Cable\_pull\_information. Each Cable\_pull\_information is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1103 Effectivity\_assignment to Class\_category\_association**

Each Effectivity\_assignment is\_applied\_to exactly one Class\_category\_association. Each Class\_category\_association is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1104 Effectivity\_assignment to Class\_condition\_association**

Each Effectivity\_assignment is\_applied\_to exactly one Class\_condition\_association. Each Class\_condition\_association is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1105 Effectivity\_assignment to Class\_inclusion\_association**

Each Effectivity\_assignment is\_applied\_to exactly one Class\_inclusion\_association. Each Class\_inclusion\_association is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1106 Effectivity\_assignment to Class\_specification\_association**

Each Effectivity\_assignment is\_applied\_to exactly one Class\_specification\_association. Each Class\_specification\_association is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1107 Effectivity\_assignment to Class\_structure\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Class\_structure\_relationship. Each Class\_structure\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1108 Effectivity\_assignment to Classification\_system**

Each Effectivity\_assignment is\_applied\_to exactly one Classification\_system. Each Classification\_system is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1109 Effectivity\_assignment to Complex\_product**

Each Effectivity\_assignment is\_applied\_to exactly one Complex\_product. Each Complex\_product is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1110 Effectivity\_assignment to Complex\_product\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Complex\_product\_relationship. Each Complex\_product\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1111 Effectivity\_assignment to Composition\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Composition\_relationship. Each Composition\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1112 Effectivity\_assignment to Configuration**

Each Effectivity\_assignment is\_applied\_to exactly one Configuration. Each Configuration is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1113 Effectivity\_assignment to Connectivity\_definition**

Each Effectivity\_assignment is\_applied\_to exactly one Connectivity\_definition. Each Connectivity\_definition is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1114 Effectivity\_assignment to Connectivity\_definition\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Connectivity\_definition\_relationship. Each Connectivity\_definition\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1115 Effectivity\_assignment to Data\_element**

Each Effectivity\_assignment is\_applied\_to exactly one Data\_element. Each Data\_element is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1116 Effectivity\_assignment to Data\_element\_association**

Each Effectivity\_assignment is\_applied\_to exactly one Data\_element\_association. Each Data\_element\_association is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1117 Effectivity\_assignment to Data\_element\_definition**

Each Effectivity\_assignment is\_applied\_to exactly one Data\_element\_definition. Each Data\_element\_definition is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1118 Effectivity\_assignment to Data\_element\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Data\_element\_relationship. Each Data\_element\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1119 Effectivity\_assignment to Design\_discipline\_item\_definition**

Each Effectivity\_assignment is\_applied\_to exactly one Design\_discipline\_item\_definition. Each Design\_discipline\_item\_definition is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1120 Effectivity\_assignment to Device**

Each Effectivity\_assignment is\_applied\_to exactly one Device. Each Device is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1121 Effectivity\_assignment to Device\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Device\_relationship. Each Device\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1122 Effectivity\_assignment to Document**

Each Effectivity\_assignment is\_applied\_to exactly one Document. Each Document is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1123 Effectivity\_assignment to Document\_file**

Each Effectivity\_assignment is\_applied\_to exactly one Document\_file. Each Document\_file is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1124 Effectivity\_assignment to Document\_file\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Document\_file\_relationship. Each Document\_file\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1125 Effectivity\_assignment to Document\_representation**

Each Effectivity\_assignment is\_applied\_to exactly one Document\_representation. Each Document\_representation is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1126 Effectivity\_assignment to Document\_version**

Each Effectivity\_assignment is\_applied\_to exactly one Document\_version. Each Document\_version is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1127 Effectivity\_assignment to Document\_version\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Document\_version\_relationship. Each Document\_version\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1128 Effectivity\_assignment to Drawing**

Each Effectivity\_assignment is\_applied\_to exactly one Drawing. Each Drawing is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1129 Effectivity\_assignment to Drawing\_sequence**

Each Effectivity\_assignment is\_applied\_to exactly one Drawing\_sequence. Each Drawing\_sequence is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1130 Effectivity\_assignment to Drawing\_sheet**

Each Effectivity\_assignment is\_applied\_to exactly one Drawing\_sheet. Each Drawing\_sheet is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1131 Effectivity\_assignment to Drawing\_sheet\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Drawing\_sheet\_relationship. Each Drawing\_sheet\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1132 Effectivity\_assignment to Effectivity**

Each Effectivity\_assignment refers to exactly one Effectivity in the role of assigned\_effectivity. Each Effectivity acts as assigned\_effectivity for zero, one, or more Effectivity\_assignment objects.

#### **4.3.1133 Effectivity\_assignment to Function\_definition**

Each Effectivity\_assignment is\_applied\_to exactly one Function\_definition. Each Function\_definition is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1134 Effectivity\_assignment to Function\_definition\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Function\_definition\_relationship. Each Function\_definition\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1135 Effectivity\_assignment to Function\_interface**

Each Effectivity\_assignment is\_applied\_to exactly one Function\_interface. Each Function\_interface is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1136 Effectivity\_assignment to Function\_unit**

Each Effectivity\_assignment is\_applied\_to exactly one Function\_unit. Each Function\_unit is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1137 Effectivity\_assignment to Function\_unit\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Function\_unit\_relationship. Each Function\_unit\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1138 Effectivity\_assignment to Function\_version**

Each Effectivity\_assignment is\_applied\_to exactly one Function\_version. Each Function\_version is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1139 Effectivity\_assignment to Function\_version\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Function\_version\_relationship. Each Function\_version\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1140 Effectivity\_assignment to Functional\_connectivity\_definition**

Each Effectivity\_assignment is\_applied\_to exactly one Functional\_connectivity\_definition. Each Functional\_connectivity\_definition is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1141 Effectivity\_assignment to Functional\_connectivity\_definition\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Functional\_connectivity\_definition\_relationship. Each Functional\_connectivity\_definition\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1142 Effectivity\_assignment to Generic\_note**

Each Effectivity\_assignment is\_applied\_to exactly one Generic\_note. Each Generic\_note is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1143 Effectivity\_assignment to Interface**

Each Effectivity\_assignment is\_applied\_to exactly one Interface. Each Interface is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1144 Effectivity\_assignment to Interface\_port**

Each Effectivity\_assignment is\_applied\_to exactly one Interface\_port. Each Interface\_port is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1145 Effectivity\_assignment to Interface\_terminal**

Each Effectivity\_assignment is\_applied\_to exactly one Interface\_terminal. Each Interface\_terminal is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1146 Effectivity\_assignment to Item**

Each Effectivity\_assignment is\_applied\_to exactly one Item. Each Item is related to zero, one, or more Effectivity\_assignment objects.



#### **4.3.1147 Effectivity\_assignment to Item\_definition\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Item\_definition\_relationship. Each Item\_definition\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1148 Effectivity\_assignment to Item\_version**

Each Effectivity\_assignment is\_applied\_to exactly one Item\_version. Each Item\_version is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1149 Effectivity\_assignment to Item\_version\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Item\_version\_relationship. Each Item\_version\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1150 Effectivity\_assignment to Location**

Each Effectivity\_assignment is\_applied\_to exactly one Location. Each Location is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1151 Effectivity\_assignment to Location\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Location\_relationship. Each Location\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1152 Effectivity\_assignment to Marking**

Each Effectivity\_assignment is\_applied\_to exactly one Marking. Each Marking is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1153 Effectivity\_assignment to Node**

Each Effectivity\_assignment is\_applied\_to exactly one Node. Each Node is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1154 Effectivity\_assignment to Node\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Node\_relationship. Each Node\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1155 Effectivity\_assignment to Notification**

Each Effectivity\_assignment is\_applied\_to exactly one Notification. Each Notification is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1156 Effectivity\_assignment to Notification\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Notification\_relationship. Each Notification\_relationship is related to zero, one, or more Effectivity\_assignment objects.



**4.3.1157 Effectivity\_assignment to Path**

Each Effectivity\_assignment is\_applied\_to exactly one Path. Each Path is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1158 Effectivity\_assignment to Path\_node**

Each Effectivity\_assignment is\_applied\_to exactly one Path\_node. Each Path\_node is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1159 Effectivity\_assignment to Path\_node\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Path\_node\_relationship. Each Path\_node\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1160 Effectivity\_assignment to Path\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Path\_relationship. Each Path\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1161 Effectivity\_assignment to Physical\_assembly\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Physical\_assembly\_relationship. Each Physical\_assembly\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1162 Effectivity\_assignment to Physical\_instance**

Each Effectivity\_assignment is\_applied\_to exactly one Physical\_instance. Each Physical\_instance is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1163 Effectivity\_assignment to Port**

Each Effectivity\_assignment is\_applied\_to exactly one Port. Each Port is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1164 Effectivity\_assignment to Process\_variable**

Each Effectivity\_assignment is\_applied\_to exactly one Process\_variable. Each Process\_variable is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1165 Effectivity\_assignment to Process\_variable\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Process\_variable\_relationship. Each Process\_variable\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1166 Effectivity\_assignment to Product\_class**

Each Effectivity\_assignment is\_applied\_to exactly one Product\_class. Each Product\_class is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1167 Effectivity\_assignment to Product\_identification**

Each Effectivity\_assignment is\_applied\_to exactly one Product\_identification. Each Product\_identification is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1168 Effectivity\_assignment to Product\_structure\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Product\_structure\_relationship. Each Product\_structure\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1169 Effectivity\_assignment to Requirement**

Each Effectivity\_assignment is\_applied\_to exactly one Requirement. Each Requirement is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1170 Effectivity\_assignment to Route**

Each Effectivity\_assignment is\_applied\_to exactly one Route. Each Route is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1171 Effectivity\_assignment to Route\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Route\_relationship. Each Route\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1172 Effectivity\_assignment to Section**

Each Effectivity\_assignment is\_applied\_to exactly one Section. Each Section is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1173 Effectivity\_assignment to Section\_end**

Each Effectivity\_assignment is\_applied\_to exactly one Section\_end. Each Section\_end is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1174 Effectivity\_assignment to Section\_interface**

Each Effectivity\_assignment is\_applied\_to exactly one Section\_interface. Each Section\_interface is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1175 Effectivity\_assignment to Section\_interface\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Section\_interface\_relationship. Each Section\_interface\_relationship is related to zero, one, or more Effectivity\_assignment objects.

#### **4.3.1176 Effectivity\_assignment to Section\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Section\_relationship. Each Section\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1177 Effectivity\_assignment to Security\_classification**

Each Effectivity\_assignment is\_applied\_to exactly one Security\_classification. Each Security\_classification is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1178 Effectivity\_assignment to Signal**

Each Effectivity\_assignment is\_applied\_to exactly one Signal. Each Signal is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1179 Effectivity\_assignment to Signal\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Signal\_relationship. Each Signal\_relationship is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1180 Effectivity\_assignment to Signal\_value**

Each Effectivity\_assignment is\_applied\_to exactly one Signal\_value. Each Signal\_value is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1181 Effectivity\_assignment to Specification**

Each Effectivity\_assignment is\_applied\_to exactly one Specification. Each Specification is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1182 Effectivity\_assignment to Specification\_category**

Each Effectivity\_assignment is\_applied\_to exactly one Specification\_category. Each Specification\_category is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1183 Effectivity\_assignment to Specification\_expression**

Each Effectivity\_assignment is\_applied\_to exactly one Specification\_expression. Each Specification\_expression is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1184 Effectivity\_assignment to Specification\_inclusion**

Each Effectivity\_assignment is\_applied\_to exactly one Specification\_inclusion. Each Specification\_inclusion is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1185 Effectivity\_assignment to Technical\_system**

Each Effectivity\_assignment is\_applied\_to exactly one Technical\_system. Each Technical\_system is related to zero, one, or more Effectivity\_assignment objects.

**4.3.1186 Effectivity\_assignment to Technical\_system\_relationship**

Each Effectivity\_assignment is\_applied\_to exactly one Technical\_system\_relationship. Each Technical\_system\_relationship is related to zero, one, or more Effectivity\_assignment objects.