



UPDM 2 META MODEL

18.3
user guide

No Magic, Inc.
2016

All material contained herein is considered proprietary information owned by No Magic, Inc. and is not to be shared, copied, or reproduced by any means. All information copyright 2009-2016 by No Magic, Inc. All Rights Reserved.

INTRODUCTION

This document presents the MagicDraw UPDM Profile structure and its representation in MagicDraw. For more information about UPDM, see the latest UPDM specification at <http://www.omg.org/spec/UPDM/>.

The MagicDraw UPDM Profile document lists MagicDraw UPDM Profile elements in alphabetical order. The element description includes table with the following columns: attribute name, attribute type, attribute owner and sample template expression (VTL).

See the sample of the table below.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
allocatedFrom	NamedElement	Allocated	\$Allocated[i].allocatedFrom
allocatedTo	NamedElement	Allocated	\$Allocated[i].allocatedTo

Table – sample of MagicDraw UPDM Profile element description

Attribute Name

The Attribute Name column provides name of property used in the MagicDraw UPDM Profile.

Attribute Type

The Attribute Type column provides name of property's type (another MagicDraw UPDM Profile element).

Attribute Owner

The Attribute Owner column provides name of property's owner in model hierarchy. Some elements properties are derived from super elements.

Sample Template Expression (VTL) for reports generation

Sample Template Expression (VTL) is the last column from the table, which gives the expression for reports generation. This expression allows to print value of the element's attribute in a report. For more information about VTL code, please see "MagicDraw Report Wizard UserGuide.pdf", "Template Variables" section.

UPDM META MODEL

Table of Contents

1.	A1	33
2.	A1 PACKAGE	33
3.	A1 REPORT	33
4.	A5	33
5.	A5 PACKAGE	33
6.	A6	33
7.	A6 PACKAGE	34
8.	A7	34
9.	A7 PACKAGE	34
10.	A8	34
11.	A8 FORECAST	34
12.	A8 PACKAGE	35
13.	ABSTRACTREFERENCEPROPERTY	35
14.	ACCEPTCHANGESTRUCTURALFEATUREEVENTACTION	36
15.	ACQUISITION VIEWPOINT	36
16.	ACTIVEVALIDATIONSUITE	36
17.	ACTIVITY	36
18.	ACTIVITYPARTOFCAPABILITY	36
19.	ACTIVITYPARTOFPROJECT	37
20.	ACTIVITYPERFORMEDBYPERFORMER	37
21.	ACTIVITYSUBJECT	38
22.	ACTUALDODAFPROPERTIES	38
23.	ACTUALLOCATION	38
24.	ACTUALLOCATIONCONCEPTROLE	39
25.	ACTUALMEASUREMENT	40
26.	ACTUALORGANIZATION	40
27.	ACTUALORGANIZATIONALRESOURCE	41
28.	ACTUALORGANIZATIONRELATIONSHIP	41
29.	ACTUALORGANIZATIONROLE	42
30.	ACTUALPERSON	43
31.	ACTUALPOST	43

32.	ACTUALPROJECT.....	44
33.	ACTUALPROJECTMILESTONE.....	44
34.	ACTUALPROJECTMILESTONEROLE.....	45
35.	ACTUALPROPERTY.....	45
36.	ACTUALPROPERTYSET.....	46
37.	ACTUALPROPERTYSETKIND.....	46
38.	ACTUATOR.....	47
39.	AcV-1.....	47
40.	AcV-1 PACKAGE.....	47
41.	AcV-2.....	48
42.	AcV-2 PACKAGE.....	48
43.	ADJUNCTPROPERTY.....	48
44.	AFCONVERT.....	48
45.	AGENT.....	48
46.	AGGREGATIONKIND.....	49
47.	ALIAS.....	49
48.	ALL VIEWS REPORT.....	49
49.	ALL VIEWS VIEWPOINT.....	49
50.	ALLOCATE.....	50
51.	ALLOCATEACTIVITYPARTITION.....	50
52.	ALLOCATED.....	50
53.	APPROVALSTATUS.....	50
54.	ARBITRARYCONNECTOR.....	51
55.	ARBITRARYRELATIONSHIPDASHEDLEFT.....	51
56.	ARBITRARYRELATIONSHIPDASHEDRIGHT.....	51
57.	ARBITRARYRELATIONSHIPDIRECTEDLEFT.....	52
58.	ARBITRARYRELATIONSHIPDIRECTEDRIGHT.....	52
59.	ARBITRARYRELATIONSHIPSOLID.....	52
60.	ARCHITECTURALDESCRIPTION.....	52
61.	ARCHITECTURALREFERENCE.....	53
62.	ARCHITECTURE META-DATA VIEWPOINT.....	53
63.	ARCHITECTUREFRAMEWORKKIND.....	54
64.	ARCHITECTUREINTRODUCTION.....	54
65.	ARCHITECTUREMETADATA.....	54
66.	ASSEMBLY.....	55
67.	ASSOCIATIONOFINFORMATION.....	55
68.	ASYNCHRONOUSMESSAGE.....	55
69.	ATTACHEDFILE.....	56

70.	ATTACHMENT	56
71.	AUTOGENERATEDNAME	56
72.	AUTOIMAGE SIZE	56
73.	AUTONUMBER	57
74.	AUXILIARY	57
75.	AUXILIARYRESOURCE	57
76.	AV-1	57
77.	AV-1 DARS REPORT	58
78.	AV-1 PACKAGE	58
79.	AV-1 REPORT	58
80.	AV-2	58
81.	AV-2 PACKAGE	59
82.	AV-2 REPORT	59
83.	AV-2 TABULAR REPORT	59
84.	BASICINTERVAL	59
85.	BINDINGCONNECTOR	60
86.	BLACKBOXICDTABLE	60
87.	BLOCK	60
88.	BLOCKHIERARCHY	61
89.	BLOCKPROPERTY	61
90.	BOOLEAN	61
91.	BOUNDARY	61
92.	BOUNDARY SYSTEM	61
93.	BOUNDREFERENCE	62
94.	BUILDCOMPONENT	62
95.	BUSINESSREQUIREMENT	62
96.	C1	62
97.	C1 PACKAGE	63
98.	C1-S1	63
99.	C2	63
100.	C2 PACKAGE	64
101.	C3	64
102.	C3 PACKAGE	64
103.	C4	64
104.	C4 PACKAGE	65
105.	C5	65
106.	C5 PACKAGE	65
107.	C7 ACTUAL PERFORMANCE PARAMETERS	65

108.	C7 PACKAGE	66
109.	C7 PERFORMANCE PARAMETERS.....	66
110.	C7 REPORT	66
111.	CALL	66
112.	CALLCONCURRENCYKIND	67
113.	CAPABILITY	67
114.	CAPABILITY.....	68
115.	CAPABILITY BEHAVIORAL MAP	68
116.	CAPABILITY STRUCTURAL MAP	69
117.	CAPABILITY VIEWPOINT	69
118.	CAPABILITYCONFIGURATION.....	70
119.	CAPABILITYCONFIGURATIONCONCEPTROLE	70
120.	CAPABILITYOFPERFORMER	71
121.	CAPABILITYPROPERTY.....	72
122.	CAPABLEELEMENT	72
123.	CHANGESTRUCTURALFEATUREEVENT	72
124.	CLASSIFICATIONTYPE.....	73
125.	CLASSIFIERBEHAVIORPROPERTY	73
126.	CLIMATE	73
127.	COLLABORATION	74
128.	COLORHOLDER	74
129.	COMMAND	74
130.	COMPETENCE.....	75
131.	COMPETENCEPROVIDER.....	76
132.	COMPETENCEREQUIRER.....	76
133.	COMPLETIONSTATUS	76
134.	COMPLEX	77
135.	COMPOSITE	77
136.	CONCEPTITEM	77
137.	CONCEPTROLE.....	78
138.	CONCEPTS VIEWPOINT.....	78
139.	CONCEPTUAL VIEW	78
140.	CONDITION.....	78
141.	CONDITIONPROPERTY.....	79
142.	CONDITIONTYPE	80
143.	CONFORM.....	80
144.	CONNECTORKIND	80
145.	CONNECTORPROPERTY	80

146.	CONSTRAINTBLOCK	81
147.	CONSTRAINTPARAMETER.....	81
148.	CONSTRAINTPROPERTY	81
149.	CONSTRUCTOR	81
150.	CONSUMER.....	82
151.	CONTENT	82
152.	CONTEXTDIAGRAM	82
153.	CONTEXTSPECIFICDEFAULTVALUE	82
154.	CONTEXTSPECIFICVALUESHOLDER	82
155.	CONTINUOUS	82
156.	CONTRACT	83
157.	CONTROL	83
158.	CONTROL.....	84
159.	CONTROLOPERATOR	84
160.	CONTROLVALUE	84
161.	COPY	84
162.	CR.....	85
163.	CR PACKAGE	85
164.	CREATE.....	85
165.	CUSTOMIMAGEHOLDER	85
166.	CUSTOMIZATION	85
167.	CUSTOMIZATIONGROUPNAMES	87
168.	CV-1	87
169.	CV-1 PACKAGE.....	87
170.	CV-2	87
171.	CV-2 PACKAGE.....	87
172.	CV-3	88
173.	CV-3 PACKAGE.....	88
174.	CV-4	88
175.	CV-4 PACKAGE.....	88
176.	CV-5	88
177.	CV-5 PACKAGE.....	89
178.	CV-5 REPORT	89
179.	CV-6	89
180.	CV-6 PACKAGE.....	90
181.	CV-7	90
182.	CV-7 PACKAGE.....	90
183.	D1	91

184.	D1 PACKAGE	91
185.	D1 REPORT.....	91
186.	D1 TABULAR REPORT	92
187.	D2.....	92
188.	D2 PACKAGE	92
189.	DARS TEMPLATE.....	92
190.	DATA AND INFORMATION VIEWPOINT	95
191.	DATAMODEL	95
192.	DATA RESTRICTIONS	95
193.	DEBUGICON	96
194.	DEFINITION	96
195.	DEFINITION	96
196.	DELEGATE	96
197.	DEPENDENCYMATRIX.....	96
198.	DEPLOYED RESOURCES VIEWPOINT.....	97
199.	DEPLOYED MILESTONE.....	97
200.	DEPLOYMENT VIEW.....	98
201.	DEPRECATED	98
202.	DERIVE	98
203.	DERIVED PROPERTIES SUITE.....	99
204.	DERIVED PROPERTY SPECIFICATION	99
205.	DERIVE REQ	99
206.	DESIGN CONSTRAINT	99
207.	DESIGN MODEL	100
208.	DESIGN RULE	100
209.	DESIRED EFFECT	101
210.	DESIRED STATE	102
211.	DESIRER	102
212.	DESTROY	102
213.	DESTRUCTOR	102
214.	DETAILS	102
215.	DEVELOPMENT STATUS	103
216.	DIAGRAM DESCRIPTION	103
217.	DIAGRAM COLLECTING METHOD.....	103
218.	DIAGRAM INFO	104
219.	DIAGRAM LEGEND	104
220.	DIAGRAMS	104
221.	DIAGRAMS DEFAULT NAME.....	104

222.	DIAGRAMTABLE	105
223.	DIAGRAMUSAGE	105
224.	DIRECTEDFEATURE	105
225.	DIRECTEDRELATIONSHIPPROPERTYPATH	106
226.	DIRECTION	106
227.	DISCRETE	106
228.	DISTRIBUTEDPROPERTY	107
229.	DIV-1	107
230.	DIV-1 PACKAGE	107
231.	DIV-2	107
232.	DIV-2 PACKAGE	107
233.	DIV-3	107
234.	DIV-3 PACKAGE	107
235.	DLODSEGMENT	108
236.	DLODSTATUS	108
237.	DOCUMENT	108
238.	DOCUMENTACCESSLEVEL	109
239.	DoDAF 2.0 ALL VIEWS REPORT	109
240.	DoDAF 2.0 AV-1 REPORT	109
241.	DoDAF 2.0 AV-2 REPORT	109
242.	DoDAF 2.0 AV-2 TABULAR REPORT	109
243.	DoDAF 2.0 OV-3 REPORT	109
244.	DoDAF 2.0 OV-3 ROLE BASED REPORT	110
245.	DoDAF 2.0 OV-6A REPORT	110
246.	DoDAF 2.0 SV-10A REPORT	110
247.	DoDAF 2.0 SV-6 REPORT	110
248.	DoDAF 2.0 SV-6 ROLE BASED REPORT	110
249.	DoDAF 2.0 SV-7 REPORT	110
250.	DoDAF 2.0 SV-8 REPORT	111
251.	DoDAFPROPERTIES	111
252.	DOMAIN	111
253.	DR	111
254.	DR PACKAGE	112
255.	DR REPORT	112
256.	DRAGANDDROP SPECIFICATION	112
257.	ED	113
258.	EFFBD	113
259.	ELEMENTGROUP	113

260.	ELEMENTPROPERTYPATH	113
261.	ELEMENTSLIBRARY	114
262.	ELEMENTSLIBRARYBRANCH.....	114
263.	ENDPATHMULTIPLICITY	114
264.	ENDURINGTASK.....	114
265.	ENERGY	115
266.	ENTERPRISEGOAL	115
267.	ENTERPRISEPHASE	116
268.	ENTERPRISEVISION	117
269.	ENTITY	117
270.	ENTITYATTRIBUTE.....	117
271.	ENTITYITEM	118
272.	ENTITYRELATIONSHIP	118
273.	ENVIRONMENT.....	119
274.	ENVIRONMENTAL EFFECT.....	119
275.	ENVIRONMENTPROPERTY.....	119
276.	ER DIAGRAM.....	120
277.	ERRORICON	120
278.	ESSENTIAL	120
279.	EXCHANGE	120
280.	EXCHANGEELEMENT	121
281.	EXCHANGEELEMENTKIND	121
282.	EXECUTABLE	122
283.	EXHIBITS.....	122
284.	EXPANSIONKIND.....	122
285.	EXPOSE.....	123
286.	EXPOSE.....	123
287.	EXTENDEDREQUIREMENT.....	123
288.	EXTERNAL.....	123
289.	EXTERNAL SYSTEM	124
290.	EXTERNALINDIVIDUAL	124
291.	EXTERNALTUPLE	124
292.	EXTERNALTUPLETYPE.....	125
293.	EXTERNALTYPE	125
294.	FATALICON.....	125
295.	FEATUREDIRECTION	126
296.	FIELDDEDCAPABILITY	126
297.	FIGUREALIGNKIND	126

298.	FILE.....	127
299.	FILEVIEW.....	127
300.	FILLSPOST.....	127
301.	FINDINGS.....	127
302.	FLOWDIRECTION.....	128
303.	FLOWPORT.....	128
304.	FLOWPROPERTY.....	128
305.	FLOWSPECIFICATION.....	129
306.	FMU.....	129
307.	FOCUS.....	129
308.	FORECAST.....	129
309.	FRAMEWORK.....	130
310.	FULLPORT.....	130
311.	FUNCTION.....	130
312.	FUNCTIONACTION.....	131
313.	FUNCTIONALREQUIREMENT.....	131
314.	FUNCTIONALSTANDARD.....	132
315.	FUNCTIONEDGE.....	132
316.	GANTTCHARTDIAGRAM.....	133
317.	GEOPOLITICALEXTENT.....	133
318.	GEOPOLITICALEXTENTKIND.....	134
319.	GEOPOLITICALEXTENTTYPE.....	134
320.	GEOPOLITICALEXTENTTYPEKIND.....	135
321.	GETTER.....	135
322.	GRANULARITYLEVEL.....	135
323.	GROUPED.....	135
324.	HASGROUPNAME.....	136
325.	HIGHLEVELOPERATIONALCONCEPT.....	136
326.	HYPERLINKOWNER.....	136
327.	ICONHOLDER.....	137
328.	IDENTIFIABLEELEMENT.....	137
329.	IMAGED.....	137
330.	IMAGEFORMAT.....	137
331.	IMPLEMENT.....	138
332.	IMPLEMENTABILITY.....	138
333.	IMPLEMENTATIONMAP.....	138
334.	IMPLEMENTATIONMATRIX.....	139
335.	IMPLEMENTATIONCLASS.....	140

336.	IMPLEMENTATIONMODEL.....	140
337.	IMPLEMENTS.....	140
338.	INCREMENTMILESTONE	141
339.	INDIVIDUALPERSONROLE	141
340.	INFO	142
341.	INFOICON	142
342.	INFORMATION.....	142
343.	INFORMATIONKIND	143
344.	INSTANCETABLE	143
345.	INSTANTIATE	144
346.	INTEGER.....	144
347.	INTERACTIONOPERATORKIND	144
348.	INTERFACEBLOCK.....	144
349.	INTERFACEREQUIREMENT	145
350.	INTERVAL.....	145
351.	INVARIANT	145
352.	INVISIBLESTEREOTYPE	146
353.	INVOCATIONONNESTEDPORTACTION.....	146
354.	ISCAPABLEOFPERFORMING.....	146
355.	ISO8601DATETIME.....	147
356.	ITEMFLOW	147
357.	JOINTPOTENTIALDESIGNATOR.....	147
358.	KNOWNRESOURCE	147
359.	L1	148
360.	L1 NODE TYPES	148
361.	L1 PACKAGE.....	148
362.	L1i	148
363.	L2 PACKAGE.....	149
364.	L2i	149
365.	L3	149
366.	L3 PACKAGE.....	150
367.	L3 REPORT	150
368.	L3 ROLE BASED	150
369.	L3 ROLE BASED REPORT.....	151
370.	L4	151
371.	L4 PACKAGE.....	151
372.	L4-P4	151
373.	L5	152

374.	L5 PACKAGE.....	152
375.	L6	152
376.	L6 PACKAGE.....	152
377.	L7	152
378.	L7 PACKAGE.....	152
379.	L8	153
380.	L8 PACKAGE.....	153
381.	L8 REPORT	153
382.	LEGENDITEM	154
383.	LEGENDLOCATION	154
384.	LIBRARY	154
385.	LIGHTCONDITION.....	155
386.	LOCATION	155
387.	LOCATIONHOLDER.....	156
388.	LOCATIONKIND	156
389.	LOCATIONTYPE	156
390.	LOCATIONTYPECONCEPTROLE.....	157
391.	LOCATIONTYPEKIND.....	157
392.	LOGICAL VIEWPOINT	158
393.	LOGICALARCHITECTURE.....	158
394.	LOGICALDATAMODEL.....	158
395.	LR	159
396.	LR PACKAGE	159
397.	LR PROGRAMME TO CAPABILITY MAPPING	159
398.	LRC	160
399.	MAPSTOCAPABILITY	160
400.	MATERIEL	161
401.	MATRIXFILTER	161
402.	MEASURE.....	162
403.	MEASUREMENT.....	162
404.	MEASUREMENTSET	163
405.	MEASURETYPE	163
406.	MERGEDDIAGRAM.....	164
407.	MESSAGEKIND.....	164
408.	MESSAGESORT	164
409.	MESSAGETYPE.....	164
410.	METACLASS	165
411.	METADATA	165

412.	METAINFO.....	165
413.	METAMODEL	165
414.	METAPROPERTY	166
415.	METRICDEFINITION	166
416.	METRICINSTANCE	166
417.	METRICSUITE	166
418.	MIGRATIONLOG	166
419.	MILESTONE	166
420.	MILESTONESEQUENCE	167
421.	MISSION.....	167
422.	MODELLIBRARY	168
423.	MOE.....	168
424.	MOUNT.....	168
425.	NAF 4.0 ALL VIEWS REPORT.....	168
426.	NATO ALL VIEW VIEWPOINT.....	169
427.	NATO ALL VIEWS REPORT	169
428.	NATO CAPABILITY VIEWPOINT	169
429.	NATO OPERATIONAL VIEWPOINT	169
430.	NATO PROGRAMME VIEWPOINT.....	169
431.	NATO SERVICE-ORIENTED VIEWPOINT	169
432.	NATO SYSTEMS VIEWPOINT	170
433.	NATO TECHNICAL VIEWPOINT.....	170
434.	NAV-1.....	170
435.	NAV-1 PACKAGE	170
436.	NAV-1 REPORT	170
437.	NAV-2.....	170
438.	NAV-2 PACKAGE	171
439.	NAV-2 REPORT	171
440.	NAV-2 TABULAR REPORT.....	171
441.	NCV-1	171
442.	NCV-1 PACKAGE	172
443.	NCV-2.....	172
444.	NCV-2 PACKAGE	172
445.	NCV-3.....	172
446.	NCV-3 PACKAGE	172
447.	NCV-4.....	172
448.	NCV-4 PACKAGE	173
449.	NCV-5.....	173

450. NCV-5 PACKAGE173

451. NCV-5 REPORT.....174

452. NCV-6.....174

453. NCV-6 PACKAGE174

454. NCV-7.....175

455. NCV-7 PACKAGE175

456. NEEDLINE175

457. NESTEDCONNECTOREND176

458. NoBUFFER176

459. NODE.....176

460. NODE IMPACT ANALYSIS MAP177

461. NODE ROLE IMPACT ANALYSIS MAP178

462. NODEASSOCIATION179

463. NODECONCEPTROLE179

464. NODEOPERATION179

465. NODEPARENT180

466. NODEPORT180

467. NODEROLE.....181

468. NoLONGERUSEDMILESTONE.....181

469. NONSTREAMING.....182

470. NORMAL182

471. NOTE182

472. NOV-1.....182

473. NOV-1 PACKAGE182

474. NOV-1I.....183

475. NOV-2.....183

476. NOV-2 PACKAGE183

477. NOV-2I.....183

478. NOV-3.....183

479. NOV-3 PACKAGE184

480. NOV-3 REPORT184

481. NOV-3 ROLE BASED184

482. NOV-3 ROLE BASED REPORT185

483. NOV-4.....185

484. NOV-4 PACKAGE185

485. NOV-5.....186

486. NOV-5 PACKAGE186

487. NOV-6A.....186

488. NOV-6A PACKAGE 186

489. NOV-6A REPORT 187

490. NOV-6B 187

491. NOV-6B PACKAGE 187

492. NOV-6C 187

493. NOV-6C PACKAGE 187

494. NOV-7 187

495. NOV-7 PACKAGE 188

496. NPV-1 188

497. NPV-1 PACKAGE 188

498. NPV-1C 188

499. NPV-2 189

500. NPV-2 PACKAGE 189

501. NSOV-1 189

502. NSOV-1 PACKAGE 190

503. NSOV-2 190

504. NSOV-2 PACKAGE 190

505. NSOV-2 REPORT 190

506. NSOV-3 190

507. NSOV-3 PACKAGE 191

508. NSOV-4 191

509. NSOV-4 PACKAGE 191

510. NSOV-5 191

511. NSOV-5 PACKAGE 191

512. NSV-1 192

513. NSV-1 PACKAGE 192

514. NSV-10A 192

515. NSV-10A PACKAGE 192

516. NSV-10A REPORT 193

517. NSV-10B 193

518. NSV-10B PACKAGE 193

519. NSV-10C 193

520. NSV-10C PACKAGE 193

521. NSV-11 193

522. NSV-11 PACKAGE 194

523. NSV-12 194

524. NSV-12 PACKAGE 194

525. NSV-1I 195

526.	NSV-2	195
527.	NSV-2 PACKAGE.....	195
528.	NSV-2I	195
529.	NSV-3	195
530.	NSV-3 PACKAGE.....	196
531.	NSV-4	196
532.	NSV-4 PACKAGE.....	196
533.	NSV-5	196
534.	NSV-5 PACKAGE.....	197
535.	NSV-6	197
536.	NSV-6 PACKAGE.....	198
537.	NSV-6 REPORT	198
538.	NSV-6 ROLE BASED	198
539.	NSV-6 ROLE BASED REPORT.....	199
540.	NSV-7 ACTUAL	199
541.	NSV-7 PACKAGE.....	199
542.	NSV-7 REPORT	200
543.	NSV-7 TYPICAL	200
544.	NSV-8	200
545.	NSV-8 PACKAGE.....	200
546.	NSV-8 REPORT	201
547.	NSV-9	201
548.	NSV-9 PACKAGE.....	201
549.	NTV-1	202
550.	NTV-1 PACKAGE	202
551.	NTV-2	202
552.	NTV-2 PACKAGE	203
553.	NUMBER	203
554.	NUMBERINGScheme	203
555.	NUMBERINGSTYLE	203
556.	NUMBERINGSTYLE	204
557.	NUMBEROWNER	204
558.	NUMBEROWNER.....	204
559.	NUMBERPART.....	204
560.	OBJECTIVEFUNCTION	205
561.	OBJECTNODEORDERINGKIND.....	205
562.	OCLSTATE.....	205
563.	ONTOLOGYREFERENCE.....	205

564.	OPERATIONAL ACTIVITY IMPLEMENTATION MAP	206
565.	OPERATIONAL ACTIVITY MAP	206
566.	OPERATIONAL VIEW MODAF	207
567.	OPERATIONAL VIEWPOINT	207
568.	OPERATIONALACTION	208
569.	OPERATIONALACTIVITY	208
570.	OPERATIONALACTIVITYACTION	209
571.	OPERATIONALACTIVITYEDGE	209
572.	OPERATIONALCONSTRAINT	210
573.	OPERATIONALEVENTTRACE	210
574.	OPERATIONALEXCHANGE	211
575.	OPERATIONALEXCHANGEITEM	212
576.	OPERATIONALEXCHANGEKIND	212
577.	OPERATIONALMESSAGE	213
578.	OPERATIONALPARAMETER	213
579.	OPERATIONALSTATE	214
580.	OPERATIONALSTATEDESCRIPTION	214
581.	OPTIONAL	215
582.	ORGANIZATION	215
583.	ORGANIZATION STRUCTURE MAP	215
584.	ORGANIZATIONALPROJECTRELATIONSHIP	216
585.	ORGANIZATIONALRESOURCE	217
586.	ORGANIZATIONCONCEPTROLE	218
587.	ORGANIZATIONTYPE	218
588.	OUTOFSERVICEMILESTONE	219
589.	OV-1	219
590.	OV-1 PACKAGE	219
591.	OV-1i	219
592.	OV-2	220
593.	OV-2 PACKAGE	220
594.	OV-2 PACKAGE MODAF	220
595.	OV-2i	220
596.	OV-3	220
597.	OV-3 PACKAGE	221
598.	OV-3 REPORT	221
599.	OV-3 ROLE BASED	221
600.	OV-3 ROLE BASED REPORT	222
601.	OV-3-DoDAF2	222

602.	OV-3-DoDAF2 ROLE BASED	223
603.	OV-4	223
604.	OV-4 PACKAGE	224
605.	OV-5	224
606.	OV-5 PACKAGE	224
607.	OV-5 PACKAGE MODAF	224
608.	OV-6A	224
609.	OV-6A PACKAGE	225
610.	OV-6A REPORT	225
611.	OV-6A-DoDAF2	225
612.	OV-6B	226
613.	OV-6B PACKAGE.....	226
614.	OV-6C	226
615.	OV-6C PACKAGE.....	226
616.	OV-7	226
617.	OV-7 PACKAGE	227
618.	OV-7 PACKAGE MODAF	227
619.	OVERLAP	227
620.	OVERWRITE.....	227
621.	OWNERDISPLAYMODE	228
622.	OWNSPROCESS	228
623.	P1	228
624.	P1 ACTUAL QUALITY REQUIREMENTS	228
625.	P1 PACKAGE.....	229
626.	P1 REPORT	229
627.	P1 SERVICE PROVISION	229
628.	P1 SYSTEMS TO SYSTEMS MATRIX	230
629.	P1 TECHNOLOGY FORECAST	230
630.	P1 TYPICAL REQUIREMENTS.....	231
631.	P2 PACKAGE.....	231
632.	P2i.....	232
633.	P3	232
634.	P3 PACKAGE.....	233
635.	P3 REPORT	233
636.	P3 ROLE BASED	233
637.	P3 ROLE BASED REPORT	234
638.	P4	234
639.	P4 PACKAGE.....	234

640.	P5	234
641.	P5 PACKAGE	234
642.	P6	234
643.	P6 PACKAGE	235
644.	P7	235
645.	P7 PACKAGE	235
646.	P8	235
647.	P8 PACKAGE	236
648.	P8 REPORT	236
649.	PANEL.....	236
650.	PARAGRAPH KIND	236
651.	PARAMETERDEFINITION.....	236
652.	PARAMETERDIRECTIONKIND.....	236
653.	PARAMETEREFFECTKIND	237
654.	PARTICIPANT	237
655.	PARTICIPANT	238
656.	PARTICIPANTPROPERTY	238
657.	PARTPROPERTY	238
658.	PATTERN.....	238
659.	PERFORMANCEREQUIREMENT.....	238
660.	PERFORMER.....	239
661.	PERSON.....	239
662.	PERSONTYPE	240
663.	PHYSICAL RESOURCE VIEWPOINT	241
664.	PHYSICALARCHITECTURE	241
665.	PHYSICALDATAMODEL	241
666.	PHYSICALREQUIREMENT	242
667.	PHYSICALRESOURCE	242
668.	PLACEONPALETTEPROPERTY	243
669.	PORT	243
670.	POST.....	243
671.	POSTCONCEPTROLE	244
672.	PR	244
673.	PR PACKAGE.....	245
674.	PR REPORT	245
675.	PROBABILITY	245
676.	PROBLEM	245
677.	PROBLEMDOMAIN	245

678.	PROCESS	246
679.	PROCESS	246
680.	PROCESSVIEW	246
681.	PROFILEUPGRADEMAPPINGRULE.....	247
682.	PROFILEUPGRADETABLE	248
683.	PROJECT.....	248
684.	PROJECT SEQUENCE TYPES.....	248
685.	PROJECT VIEWPOINT	249
686.	PROJECTACTIVITY.....	249
687.	PROJECTACTIVITYACTION	249
688.	PROJECTACTIVITYEDGE.....	250
689.	PROJECTMILESTONE.....	250
690.	PROJECTMILESTONEROLE.....	251
691.	PROJECTOWNERSHIP	251
692.	PROJECTSEQUENCE	251
693.	PROJECTSTATUS	252
694.	PROJECTTHEME	253
695.	PROJECTTYPE	253
696.	PROPERTY	253
697.	PROPERTY	254
698.	PROPERTYGROUP	254
699.	PROPERTYSET.....	255
700.	PROPERTYSPECIFICTYPE.....	255
701.	PROPRIETARYINFORMATION.....	255
702.	PROTOCOL.....	255
703.	PROTOCOLIMPLEMENTATION.....	256
704.	PROTOCOLLAYER.....	256
705.	PROVIDER	257
706.	PROVIDESCOMPETENCE	257
707.	PROXYPORT	257
708.	PSEUDOSTATEKIND	258
709.	PV-1.....	258
710.	PV-1 PACKAGE	258
711.	PV-2.....	259
712.	PV-2 PACKAGE	259
713.	PV-3.....	259
714.	PV-3 PACKAGE	260
715.	QUANTITYKIND	260

716.	RATE	260
717.	RATIONALE	260
718.	REAL	261
719.	REALIZATION.....	261
720.	REALIZINGELEMENT.....	261
721.	REFERENCEPROPERTY	261
722.	REFINE.....	262
723.	REFINE.....	262
724.	RELATIONMAP.....	262
725.	RELATIONMAPLAYOUTENUMERATION	264
726.	RELATIONOPTION	265
727.	RELEASABILITY	265
728.	REPLACESTEREOTYPE	265
729.	REPLACETAGGEDVALUE	266
730.	REPLACETYPE	266
731.	REPORTCATEGORY.....	266
732.	REPORTDATA	266
733.	REPORTDATAMAPPINGRULE.....	267
734.	REPORTTEMPLATE	267
735.	REPRESENTATION KIND	267
736.	REQUEST.....	268
737.	REQUEST.....	268
738.	REQUIREMENT	269
739.	REQUIREMENTRELATED.....	269
740.	REQUIREMENTTABLE	269
741.	REQUIRESCOMPETENCE	270
742.	RESOURCE.....	271
743.	RESOURCE IMPACT ANALYSIS MAP.....	271
744.	RESOURCE ROLE IMPACT ANALYSIS MAP	272
745.	RESOURCEACTION	273
746.	RESOURCEARTIFACT	274
747.	RESOURCEARTIFACTCONCEPTROLE.....	274
748.	RESOURCEASSOCIATION	275
749.	RESOURCECONNECTOR.....	275
750.	RESOURCECONSTRAINT	275
751.	RESOURCEEVENTTRACE	276
752.	RESOURCEINTERACTION	276
753.	RESOURCEINTERACTIONITEM.....	277

754.	RESOURCEINTERFACE.....	278
755.	RESOURCEMESSAGE.....	278
756.	RESOURCEOPERATION.....	279
757.	RESOURCEPARAMETER.....	279
758.	RESOURCEPORT.....	280
759.	RESOURCEROLE.....	280
760.	RESOURCEROLEMAPPING.....	281
761.	RESOURCEROLETABLE.....	281
762.	RESOURCESTATE.....	281
763.	RESOURCESTATEMACHINE.....	281
764.	RESPONSIBILITY.....	282
765.	RESPONSIBILITY.....	282
766.	RISKKIND.....	283
767.	ROLEKIND.....	283
768.	ROLETYPE.....	283
769.	RULE.....	284
770.	RULEKIND.....	285
771.	S1.....	285
772.	S1 PACKAGE.....	285
773.	S3.....	285
774.	S3 PACKAGE.....	285
775.	S3 REPORT.....	286
776.	S4.....	286
777.	S4 PACKAGE.....	286
778.	S4 SERVICES TO OPERATIONAL ACTIVITIES MAPPING.....	286
779.	S5.....	287
780.	S5 PACKAGE.....	287
781.	S6.....	287
782.	S6 PACKAGE.....	287
783.	S7.....	287
784.	S7 PACKAGE.....	287
785.	S8.....	288
786.	S8 PACKAGE.....	288
787.	S8 REPORT.....	288
788.	SAMEAS.....	288
789.	SATISFY.....	289
790.	SCRIPT.....	289
791.	SDDSUBSYSTEM.....	289

792.	SECURITYATTRIBUTESGROUP	290
793.	SECURITYCLASSIFICATION	290
794.	SECURITYDOMAIN	291
795.	SEMANTIC	292
796.	SEMANTICATTRIBUTE.....	292
797.	SEND	292
798.	SENSOR.....	292
799.	SEQUENCETYPE	292
800.	SERVICE.....	293
801.	SERVICE.....	293
802.	SERVICE.....	293
803.	SERVICE ORIENTED VIEWPOINT	294
804.	SERVICE VIEWPOINT.....	294
805.	SERVICEACCESS	294
806.	SERVICEACTION	294
807.	SERVICEATTRIBUTE	295
808.	SERVICECHANNEL.....	295
809.	SERVICECONTRACT.....	295
810.	SERVICEDESCRIPTION.....	295
811.	SERVICEFEATURE	296
812.	SERVICEFUNCTION.....	297
813.	SERVICEFUNCTIONACTION.....	297
814.	SERVICEFUNCTIONEDGE.....	298
815.	SERVICEINTERACTION	298
816.	SERVICEINTERFACE.....	299
817.	SERVICEINTERFACE.....	299
818.	SERVICELEVEL VALUE	299
819.	SERVICELEVEL VALUESET.....	300
820.	SERVICEMESSAGE.....	300
821.	SERVICEMESSAGEHANDLER.....	301
822.	SERVICEOPERATION	301
823.	SERVICEPARAMETER	302
824.	SERVICEPOLICY	302
825.	SERVICEPORT	303
826.	SERVICES VIEWPOINT	303
827.	SERVICESARCHITECTURE.....	304
828.	SERVICESTATEMACHINE	304
829.	SETTER.....	304

830.	SEVERITYKIND	304
831.	SHAREDPROPERTY	305
832.	SKILL	305
833.	SKILLOFPERSONTYPE.....	305
834.	SMARTPACKAGE	306
835.	SOA MATRIX	306
836.	SOA SUMMARY	307
837.	SOFTWARE	307
838.	SOFTWARECONCEPTROLE.....	308
839.	SORTINGMODE	308
840.	SOURCE	309
841.	SOV-1	309
842.	SOV-1 PACKAGE.....	309
843.	SOV-2	309
844.	SOV-2 PACKAGE.....	309
845.	SOV-2 REPORT	309
846.	SOV-3	310
847.	SOV-3 PACKAGE.....	310
848.	SOV-4A	310
849.	SOV-4A PACKAGE	311
850.	SOV-4A REPORT	311
851.	SOV-4B	311
852.	SOV-4B PACKAGE.....	311
853.	SOV-4C	312
854.	SOV-4C PACKAGE.....	312
855.	SOV-5	312
856.	SOV-5 PACKAGE.....	312
857.	SPECIALHANDLINGINSTRUCTIONS	312
858.	SPECIFICATION.....	313
859.	STAKEHOLDER	314
860.	STANDARD	314
861.	STANDARDCONFIGURATION	314
862.	STANDARDOPERATIONALACTIVITY	315
863.	STANDARDS VIEWPOINT	316
864.	STATUSINDICATORS	316
865.	STDV-1	316
866.	STDV-1 PACKAGE	317
867.	STDV-2.....	317

868.	STDV-2 PACKAGE	317
869.	STEREOTYPEEXTENSION	318
870.	STEREOTYPESMAPPINGRULE.....	318
871.	STRATEGIC VIEWPOINT.....	318
872.	STREAMING	319
873.	STRING	319
874.	STRINGSMAPPINGRULE	319
875.	STRUCTURALPART	319
876.	STV-1	320
877.	STV-1 PACKAGE	320
878.	STV-2.....	320
879.	STV-2 PACKAGE	320
880.	STV-3.....	320
881.	STV-3 PACKAGE	320
882.	STV-4.....	321
883.	STV-4 PACKAGE	321
884.	STV-5.....	321
885.	STV-5 PACKAGE	321
886.	STV-5 REPORT.....	322
887.	STV-6.....	322
888.	STV-6 PACKAGE	322
889.	SUBCONTENTSKind.....	323
890.	SUBJECTOFFORECAST	323
891.	SUBJECTOFOPERATIONALCONSTRAINT	323
892.	SUBJECTOFOPERATIONALSTATEMACHINE.....	324
893.	SUBJECTOFRESOURCECONSTRAINT	324
894.	SUBSYSTEM.....	325
895.	SUBSYSTEM.....	325
896.	SUGGESTEDSTRINGVALUES.....	325
897.	SUGGESTEDVALUES.....	325
898.	SV-1.....	325
899.	SV-1 PACKAGE	326
900.	SV-1 PACKAGE MODAF.....	326
901.	SV-10A.....	326
902.	SV-10A PACKAGE	326
903.	SV-10A REPORT.....	327
904.	SV-10A-DoDAF2	327
905.	SV-10B	327

906. SV-10B PACKAGE 328

907. SV-10C 328

908. SV-10C PACKAGE 328

909. SV-11 328

910. SV-11 PACKAGE 328

911. SV-11 PACKAGE MODAF 328

912. SV-12 329

913. SV-12 PACKAGE 329

914. SV-13 329

915. SV-1I 330

916. SV-2 330

917. SV-2 PACKAGE 330

918. SV-2 PACKAGE MODAF 331

919. SV-2I 331

920. SV-3 331

921. SV-3 PACKAGE 331

922. SV-3-DoDAF2 332

923. SV-4 332

924. SV-4 PACKAGE 332

925. SV-4 PACKAGE MODAF 333

926. SV-5 333

927. SV-5 PACKAGE 333

928. SV-5A 334

929. SV-5B 334

930. SV-6 335

931. SV-6 PACKAGE 335

932. SV-6 REPORT 336

933. SV-6 ROLE BASED 336

934. SV-6 ROLE BASED REPORT 336

935. SV-6-DoDAF2 337

936. SV-6-DoDAF2 ROLE BASED 337

937. SV-7 ACTUAL 338

938. SV-7 ACTUAL-DoDAF2 338

939. SV-7 PACKAGE 339

940. SV-7 REPORT 339

941. SV-7 TYPICAL 339

942. SV-7 TYPICAL-DoDAF2 340

943. SV-8 340

944. SV-8 PACKAGE 340

945. SV-8 REPORT..... 341

946. SV-8-DoDAF2 341

947. SV-9..... 341

948. SV-9 PACKAGE 342

949. SV-9-DoDAF2 342

950. SVCV-1 342

951. SVCV-1 PACKAGE 343

952. SVCV-10A 343

953. SVCV-10A PACKAGE 343

954. SVCV-10A REPORT..... 344

955. SVCV-10B 344

956. SVCV-10B PACKAGE 344

957. SVCV-10C 344

958. SVCV-10C PACKAGE 344

959. SVCV-2..... 344

960. SVCV-2 PACKAGE 345

961. SVCV-2i..... 345

962. SVCV-3A..... 345

963. SVCV-3A PACKAGE 345

964. SVCV-3B 346

965. SVCV-3B PACKAGE 346

966. SVCV-4..... 346

967. SVCV-4 PACKAGE 346

968. SVCV-5..... 347

969. SVCV-5 PACKAGE 347

970. SVCV-6..... 347

971. SVCV-6 PACKAGE 348

972. SVCV-6 REPORT..... 348

973. SVCV-6 ROLE BASED 348

974. SVCV-6 ROLE BASED REPORT 349

975. SVCV-7 ACTUAL 349

976. SVCV-7 PACKAGE 350

977. SVCV-7 REPORT..... 350

978. SVCV-7 TYPICAL 350

979. SVCV-8..... 350

980. SVCV-8 PACKAGE 351

981. SVCV-8 REPORT..... 351

982. SvcV-9.....	351
983. SvcV-9 PACKAGE	352
984. SWIMLANEDIAGRAM.....	352
985. SYSTEM	352
986. SYSTEM	352
987. SYSTEM CONTEXT	353
988. SYSTEM PROCESS	353
989. SYSTEM RESOURCE MAP.....	353
990. SYSTEMACTION	354
991. SYSTEMCONCEPTROLE.....	354
992. SYSTEMMODEL.....	355
993. SYSTEMRESOURCE.....	355
994. SYSTEMS VIEW MODAF.....	355
995. SYSTEMS VIEWPOINT.....	356
996. SYSTEMVALIDATIONSUITE.....	356
997. TABLE.....	356
998. TABLECOLUMNNAME.....	356
999. TABLELAYOUT	357
1000. TAGGROUP	357
1001. TECHNICAL STANDARDS VIEWPOINT.....	357
1002. TECHNICALSTANDARD	357
1003. TEMPORALPART.....	358
1004. TEMPORALSCOPE	358
1005. TERM.....	359
1006. TESTCASE	359
1007. TEXTDIRECTION	359
1008. TIMELINE.....	359
1009. TIMEPERIOD	359
1010. TODO_OWNER.....	360
1011. TRACE	360
1012. TRACE	360
1013. TRANSACTIONAL.....	360
1014. TRANSACTIONALATTRIBUTE	361
1015. TRANSITIONKIND.....	361
1016. TREESTRUCTUREENUMERATION	361
1017. TRIGGERONNESTEDPORT	361
1018. TRUSTLINE	362
1019. TV-1.....	362

1020. TV-1 PACKAGE	363
1021. TV-2.....	363
1022. TV-2 PACKAGE	363
1023. TYPE.....	364
1024. TYPEMODIFIER	364
1025. TYPEMODIFIERENUMERATION	364
1026. UNIFORM	364
1027. UNIT	365
1028. UNLIMITEDNATURAL	365
1029. UPDMELEMENT	365
1030. USABILITYREQUIREMENT	365
1031. USECASEMODEL	366
1032. USECASEVIEW	366
1033. USER SYSTEM	366
1034. UTILITIES	367
1035. UTILITY	367
1036. VALIDATIONRULE	367
1037. VALIDATIONSUITE.....	367
1038. VALUEPROPERTY	367
1039. VALUETYPE	368
1040. VARIABLE	368
1041. VARIANT MAP	368
1042. VERDICTKIND.....	369
1043. VERIFICATIONMETHODKIND	369
1044. VERIFY	370
1045. VERSIONOFCONFIGURATION	370
1046. VIEW	370
1047. VIEW	371
1048. VIEW	371
1049. VIEWPOINT.....	372
1050. VIEWPOINT.....	372
1051. VIRTUAL.....	372
1052. VISIBILITYKIND	373
1053. VISION	373
1054. VISIONSTATEMENT	373
1055. WARNINGICON	374
1056. WEBREPORTNODENAME	374
1057. WHITEBOXICDTABLE	374

1058. WHOLELIFECONFIGURATION.....	375
1059. WHOLELIFEENTERPRISE	375
1060. WRAPPER	376
1061. WRAPPERATTRIBUTE	376

1. A1

Base Classifier

- [InvisibleStereotype](#)

2. A1 Package

Base Classifier

- [InvisibleStereotype](#)

3. A1 Report

Base Classifier

- [InvisibleStereotype](#)

4. A5

Base Classifier

- [InvisibleStereotype](#)

5. A5 Package

Base Classifier

- [InvisibleStereotype](#)

6. A6

Base Classifier

- [InvisibleStereotype](#)

7. A6 Package

Base Classifier

- [InvisibleStereotype](#)

8. A7

Base Classifier

- [InvisibleStereotype](#)

9. A7 Package

Base Classifier

- [InvisibleStereotype](#)

10. A8

Standards (A8) table defines the technical and non technical standards, guidance and policy applicable to the architecture. There are two ways to add a row in this table:

1. Add new UPDM Element. Click “Add new UPDM Element” button. Select UPDM Element (this term means any of the UPDM available element). Specify the owner for the selected element.

2. Add Existing UPDM elements. Click “Add Existing UPDM Element” button and select UPDM elements.

For each row there should be assigned one or more Standards or Protocols that the particular UPDM element must conform to. Click “...” button on any cell in the "Standard/Policy" column.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

11. A8 Forecast

Standards Forecast (A8) table defines expected changes in technology related standards and conventions.

You will find this table identical to the P1 Technology Forecast table. In general they are identical in implementation, but A8 Forecast is more likely to be used for Standards and Protocols forecasting.

Three major steps should be done to create the table:

1. Add Rows to the Table. *There are two ways to add a row to the table:*

1.1. Add new Subject of Forecast as row Header. *Click **Add New** button and select element you want to create. Specify owner for selected element.*

1.2. Add Existing Subject of Forecast as row Header. *Click **Add Existing** button and select one or more existing elements.*

2. Add columns to the table. *Click "**Add/Remove forecast**" button. Specify Time Periods for the forecasting: select or create Time Line Package to store Forecast dates; select forecast kind. According to the selected Forecast Kind, specify additional options needed (see Forecasting Period Dialog for more info).*

3. Fill in the cells with the Subjects of Forecast. *Click the "..." button on the cell you want to fill in, select Resources, Standards, Protocols or Competences to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected Time Period.*

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

12. A8 Package

Base Classifier

- [InvisibleStereotype](#)

13. AbstractReferenceProperty

Do not use it directly. Use either ReferenceProperty or SharedProperty instead.

Base Classifier

- [BlockProperty](#)

14. AcceptChangeStructuralFeatureEventAction

15. Acquisition Viewpoint

Base Classifier

- [InvisibleStereotype](#)

16. activeValidationSuite

Base Classifier

- [validationSuite](#)

17. Activity

UPDM: An abstract element that represents a behavior (i.e. a Function or OperationalActivity) that can be performed by a Performer.

MODAF: NA

DoDAF: Work, not specific to a single organization, weapon system or individual that transforms inputs (Resources) into outputs (Resources) or changes their state.

Base Classifier

- [Desirer](#)
- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
activityPerformableUnderCondition	Environment	Activity	<code>\$Activity[i].activityPerformableUnderCondition</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Activity[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Activity[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Activity[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Activity[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Activity[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Activity[i].URI</code>

18. ActivityPartOfCapability

Base Classifier

- [MapsToCapability](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActivityPartOfCapability[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActivityPartOfCapability[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActivityPartOfCapability[i].endBoundaryType</code>
MapsToCapability.client		MapsToCapability	<code>\$ActivityPartOfCapability[i].MapsToCapability.client</code>
MapsToCapability.supplier		MapsToCapability	<code>\$ActivityPartOfCapability[i].MapsToCapability.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$ActivityPartOfCapability[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActivityPartOfCapability[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActivityPartOfCapability[i].URI</code>

19. ActivityPartOfProject**Base Classifier**

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActivityPartOfProject[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActivityPartOfProject[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActivityPartOfProject[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ActivityPartOfProject[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActivityPartOfProject[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActivityPartOfProject[i].URI</code>

20. ActivityPerformedByPerformer

UPDM: Links a Performer to the behavior that it can perform

MODAF: NA

DoDAF: An overlap of an Activity with a Resource, in particular a consuming or producing Activity that expresses an input, output, consumption, or production Activity of the Resource

Base Classifier

- [IsCapableOfPerforming](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ActivityPerformedByPerformer[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ActivityPerformedByPerformer[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ActivityPerformedByPerformer[i].endBoundaryType
Performs.client		IsCapableOfPerforming	\$ActivityPerformedByPerformer[i].Performs.client
Performs.supplier		IsCapableOfPerforming	\$ActivityPerformedByPerformer[i].Performs.supplier
propertySet	PropertySet	UPDMElement	\$ActivityPerformedByPerformer[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ActivityPerformedByPerformer[i].startBoundaryType
URI	String	UPDMElement	\$ActivityPerformedByPerformer[i].URI

21. ActivitySubject

MODAF: Anything that is acted upon by an OperationalActivity

DoDAF: NA

Base Classifier

- UPDMElement

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	\$ActivitySubject[i].actsUpon
actualPropertySet	ActualPropertySet	UPDMElement	\$ActivitySubject[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ActivitySubject[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ActivitySubject[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ActivitySubject[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ActivitySubject[i].startBoundaryType
URI	String	UPDMElement	\$ActivitySubject[i].URI

22. ActualDoDAFProperties

23. ActualLocation

MODAF: A PhysicalLocation (MODAF::ActualLocation) is a location anywhere on the earth. The means of describing the location is a string (locationDescription). The information contained in that string is governed by the taxonomy reference - e.g. if the PhysicalLocation is a "GPS reference", the string will contain the GPS

coordinates. *NOTE: this has been extended in UPDM to include non-earth locations.*
DoDAF: All subtypes of << IndividualType>> Location, such as Facility, Site, etc.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualLocation[i].actualPropertySet</code>
address	String	ActualLocation	<code>\$ActualLocation[i].address</code>
conformsTo	Standard	UPDMElement	<code>\$ActualLocation[i].conformsTo</code>
customKind	String	ActualLocation	<code>\$ActualLocation[i].customKind</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualLocation[i].endBoundaryType</code>
locationKind	LocationKind	ActualLocation	<code>\$ActualLocation[i].locationKind</code>
locationNamedByAddress	Boolean	ActualLocation	<code>\$ActualLocation[i].locationNamedByAddress</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualLocation[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualLocation[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualLocation[i].URI</code>

24. ActualLocationConceptRole

Base Classifier

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualLocationConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$ActualLocationConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$ActualLocationConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualLocationConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualLocationConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualLocationConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualLocationConceptRole[i].URI</code>

25. ActualMeasurement

UPDM: An actual value of the Measurement.

MODAF: NA

DoDAF: NA

Base Classifier

- [ActualProperty](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualMeasurement.definingFeature		ActualMeasurement	<code>\$ActualMeasurement[i].ActualMeasurement.definingFeature</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualMeasurement[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActualMeasurement[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualMeasurement[i].endBoundaryType</code>
endDate	ISO8601DateTime	ActualProperty	<code>\$ActualMeasurement[i].endDate</code>
intention	ActualPropertySetKind	ActualProperty	<code>\$ActualMeasurement[i].intention</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualMeasurement[i].propertySet</code>
PropertyValue.definingFeature		ActualProperty	<code>\$ActualMeasurement[i].PropertyValue.definingFeature</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualMeasurement[i].startBoundaryType</code>
startDate	ISO8601DateTime	ActualProperty	<code>\$ActualMeasurement[i].startDate</code>
URI	String	UPDMElement	<code>\$ActualMeasurement[i].URI</code>

26. ActualOrganization

MODAF: An actual specific organisation, an instance of an organisation class - e.g. "The US Department of Defense"

DoDAF: [[DoDAF::Organization](#)]: A specific real-world assemblage of people and other resources organized for an on-going purpose.

Base Classifier

- [ActualOrganizationalResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualOrganization.classifier		ActualOrganization	<code>\$ActualOrganization[i].ActualOrganization.classifier</code>
ActualOrganization.slot		ActualOrganization	<code>\$ActualOrganization[i].ActualOrganization.slot</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualOrganization[i].actualPropertySet</code>
code/symbol	String	ActualOrganization	<code>\$ActualOrganization[i].code/symbol</code>
conformsTo	Standard	UPDMElement	<code>\$ActualOrganization[i].conformsTo</code>

endBoundaryType	ISO8601DateTime	UPDMElement	\$ActualOrganization[i].endBoundaryType
physicalLocation	ActualLocation	LocationHolder	\$ActualOrganization[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$ActualOrganization[i].propertySet
ratifiedStandards	Standard	ActualOrganization	\$ActualOrganization[i].ratifiedStandards
requiredEnvironment	Environment	LocationHolder	\$ActualOrganization[i].requiredEnvironment
serviceType	String	ActualOrganization	\$ActualOrganization[i].serviceType
startBoundaryType	ISO8601DateTime	UPDMElement	\$ActualOrganization[i].startBoundaryType
URI	String	UPDMElement	\$ActualOrganization[i].URI

27. ActualOrganizationalResource

UPDM: An ActualOrganization or an ActualPost.

MODAF: An instance of either an actual organisation or an actual post.

DoDAF: A specific real-world assemblage of people and other resources organized for an on-going purpose.

Base Classifier

- [CompetenceProvider](#)
- [LocationHolder](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ActualOrganizationalResource[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ActualOrganizationalResource[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ActualOrganizationalResource[i].endBoundaryType
physicalLocation	ActualLocation	LocationHolder	\$ActualOrganizationalResource[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$ActualOrganizationalResource[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$ActualOrganizationalResource[i].requiredEnvironment
startBoundaryType	ISO8601DateTime	UPDMElement	\$ActualOrganizationalResource[i].startBoundaryType
URI	String	UPDMElement	\$ActualOrganizationalResource[i].URI

28. ActualOrganizationRelationship

UPDM: A relationship between two ActualOrganizationResources.

MODAF: A relationship between two actual specific organisations or parts of an organisation.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualOrganizationRelationship.clone		ActualOrganizationRelationship	\$ActualOrganizationRelationship[i].ActualOrganizationRelationship.clone
ActualOrganizationRelationship.realizes		ActualOrganizationRelationship	\$ActualOrganizationRelationship[i].ActualOrganizationRelationship.realizes
ActualOrganizationRelationship.source		ActualOrganizationRelationship	\$ActualOrganizationRelationship[i].ActualOrganizationRelationship.source
ActualOrganizationRelationship.target		ActualOrganizationRelationship	\$ActualOrganizationRelationship[i].ActualOrganizationRelationship.target
actualPropertySet	ActualPropertySet	UPDMElement	\$ActualOrganizationRelationship[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ActualOrganizationRelationship[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ActualOrganizationRelationship[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ActualOrganizationRelationship[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ActualOrganizationRelationship[i].startBoundaryType
URI	String	UPDMElement	\$ActualOrganizationRelationship[i].URI

29. ActualOrganizationRole

UPDM: Relates an actual specific organization to an actual specific organizational resource that fulfils a role in that organization.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualOrganizationPart.definingFeature		ActualOrganizationRole	\$ActualOrganizationRole[i].ActualOrganizationPart.definingFeature
ActualOrganizationPart.owningInstance		ActualOrganizationRole	\$ActualOrganizationRole[i].ActualOrganizationPart.owningInstance
actualPropertySet	ActualPropertySet	UPDMElement	\$ActualOrganizationRole[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ActualOrganizationRole[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ActualOrganizationRole[i].endBoundaryType

propertySet	PropertySet	UPDMElement	<code>\$ActualOrganizationRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualOrganizationRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualOrganizationRole[i].URI</code>

30. ActualPerson

UPDM: Named individual that fulfills an ActualPost. An individual human being (vs Person which is a type), that is recognized by law as the subject of rights and duties.

MODAF: NA

DoDAF: An individual person

Base Classifier

- [CompetenceProvider](#)
- [LocationHolder](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualPerson.classifier		ActualPerson	<code>\$ActualPerson[i].ActualPerson.classifier</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualPerson[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActualPerson[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualPerson[i].endBoundaryType</code>
filledPost	ActualPost	ActualPerson	<code>\$ActualPerson[i].filledPost</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$ActualPerson[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualPerson[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$ActualPerson[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualPerson[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualPerson[i].URI</code>

31. ActualPost

UPDM: An actual, specific post, an instance of a PostType class - e.g. "President of the United States of America."

MODAF: NA

DoDAF: NA

Base Classifier

- [ActualOrganizationalResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualPost.classifier		ActualPost	<code>\$ActualPost[i].ActualPost.classifier</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualPost[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActualPost[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualPost[i].endBoundaryType</code>
filledBy	ActualPerson	ActualPost	<code>\$ActualPost[i].filledBy</code>

physicalLocation	ActualLocation	LocationHolder	<code>\$ActualPost[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualPost[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$ActualPost[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualPost[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualPost[i].URI</code>

32. ActualProject

MODAF: (MODAF::Project): A time-limited endeavour to create a specific set of products or services.

DoDAF: (DoDAF::Project): A temporary endeavor undertaken to create Resources or Desired Effects.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualProject.classifier		ActualProject	<code>\$ActualProject[i].ActualProject.classifier</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualProject[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActualProject[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProject[i].endBoundaryType</code>
endDate	ISO8601DateTime	ActualProject	<code>\$ActualProject[i].endDate</code>
ownedMilestones	ActualProjectMilestone	ActualProject	<code>\$ActualProject[i].ownedMilestones</code>
part	ActualProject	ActualProject	<code>\$ActualProject[i].part</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualProject[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProject[i].startBoundaryType</code>
startDate	ISO8601DateTime	ActualProject	<code>\$ActualProject[i].startDate</code>
URI	String	UPDMElement	<code>\$ActualProject[i].URI</code>
whole	ActualProject	ActualProject	<code>\$ActualProject[i].whole</code>

33. ActualProjectMilestone

MODAF: (ProjectMilestone): An event in a ActualProject (MODAF::Project) by which progress is measured. Note: in the case of an acquisition project, there are two key types of milestones which shall be represented using subtypes - IncrementMilestone (MODAF::CapabilityIncrement) and OutOfServiceMilestone (MODAF::OutOfService)

DoDAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualProjectMilestone.classifier		ActualProjectMilestone	<code>\$ActualProjectMilestone[i].ActualProjectMilestone.classifier</code>
ActualProjectMilestone.slot		ActualProjectMilestone	<code>\$ActualProjectMilestone[i].ActualProjectMilestone.slot</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualProjectMilestone[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActualProjectMilestone[i].conformsTo</code>
date	ISO8601DateTime	ActualProjectMilestone	<code>\$ActualProjectMilestone[i].date</code>
description	String	ActualProjectMilestone	<code>\$ActualProjectMilestone[i].description</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProjectMilestone[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualProjectMilestone[i].propertySet</code>
resource	SystemResource	ActualProjectMilestone	<code>\$ActualProjectMilestone[i].resource</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProjectMilestone[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualProjectMilestone[i].URI</code>

34. ActualProjectMilestoneRole

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualProjectMilestoneRole[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActualProjectMilestoneRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProjectMilestoneRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualProjectMilestoneRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProjectMilestoneRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualProjectMilestoneRole[i].URI</code>

35. ActualProperty

UPDM: The value of a Measure.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualProperty[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ActualProperty[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProperty[i].endBoundaryType</code>
endDate	ISO8601DateTime	ActualProperty	<code>\$ActualProperty[i].endDate</code>
intention	ActualPropertySetKind	ActualProperty	<code>\$ActualProperty[i].intention</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualProperty[i].propertySet</code>
PropertyValue.definingFeature		ActualProperty	<code>\$ActualProperty[i].PropertyValue.definingFeature</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualProperty[i].startBoundaryType</code>
startDate	ISO8601DateTime	ActualProperty	<code>\$ActualProperty[i].startDate</code>
URI	String	UPDMElement	<code>\$ActualProperty[i].URI</code>

36. ActualPropertySet

UPDM: A set or collection of ActualMeasurement(s). A date of measurement can be set. An intent of ActualMeasurementSet can be “Result”, “Required”, or “Estimate”

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ActualPropertySet[i].actualPropertySet</code>
ActualPropertySet.classifier		ActualPropertySet	<code>\$ActualPropertySet[i].ActualPropertySet.classifier</code>
ActualPropertySet.slot		ActualPropertySet	<code>\$ActualPropertySet[i].ActualPropertySet.slot</code>
appliesTo	UPDMElement	ActualPropertySet	<code>\$ActualPropertySet[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$ActualPropertySet[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualPropertySet[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ActualPropertySet[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ActualPropertySet[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ActualPropertySet[i].URI</code>

37. ActualPropertySetKind

Possible kinds of ActualMeasurementSet intention.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

Actual	Enumeration Literal	ActualPropertySetKind	\$ActualPropertySetKind[i].Actual
Estimate	Enumeration Literal	ActualPropertySetKind	\$ActualPropertySetKind[i].Estimate
Required	Enumeration Literal	ActualPropertySetKind	\$ActualPropertySetKind[i].Required

38. Actuator

An Actuator is a special external system that influences the environment of the system under development. For example a Heater assembly or a Central locking system of a car.

Base Classifier

- [External system](#)

39. AcV-1

*The **Responsibility Matrix (AcV-1)** describes the mapping between the Actual Projects and the Actual Organizational Resources.*

The Rows of this matrix are Actual Projects and the Columns are Actual Organizational Resources (Actual Organization or Actual Post).

To build the Matrix:

- 1. Specify **Rows** scope (Actual Projects);*
- 2. Specify **Columns** scope (Actual Organizations and Actual Posts);*
- 3. Click "**Refresh**" button.*

Actual Organizational Resources maps to Actual Projects using "Organizational Project Relationship".

To map Actual Project to Actual Organizational Resource, double click on the intersection between the desired elements. By double clicking on the intersection one more time, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

40. AcV-1 Package

Base Classifier

- [InvisibleStereotype](#)

41. AcV-2

Base Classifier

- [InvisibleStereotype](#)

42. AcV-2 Package

Base Classifier

- [InvisibleStereotype](#)

43. AdjunctProperty

The *AdjunctProperty* stereotype can be applied to properties to constrain their values to the values of connectors typed by association blocks, call actions, object nodes, variables, or parameters, interaction uses, and submachine states. The values of connectors typed by association blocks are the instances of the association block typing a connector in the block having the stereotyped property. The values of call actions are the executions of behaviors invoked by the behavior having the call action and the stereotyped property (see Subclause 11.3.1.1.1 for more about this use of the stereotype). The values of object nodes are the values of tokens in the object nodes of the behavior having the stereotyped property (see Subclause 11.3.1.4.1 for more about this use of the stereotype). The values of variables are those assigned by executions of activities that have the stereotyped property. The values of parameters are those assigned by executions of behaviors that have the stereotyped property. The keyword «adjunct» before a property name indicates the property is stereotyped by *AdjunctProperty*.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
principal	Element	AdjunctProperty	<code>\$AdjunctProperty[i].principal</code>

44. AFConvert

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
stereotype	Stereotype	AFConvert	<code>\$AFConvert[i].stereotype</code>
string	String	AFConvert	<code>\$AFConvert[i].string</code>

45. Agent

An *Agent* is a classification of autonomous entities that can adapt to and interact with their environment. It describes a set of agent instances that have features, constraints, and semantics in common. Agents in SoaML are also participants, providing and using services.

Base Classifier

- [Participant](#)

46. AggregationKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
composite	Enumeration Literal	AggregationKind	<code>\$AggregationKind[i].composite</code>
none	Enumeration Literal	AggregationKind	<code>\$AggregationKind[i].none</code>
shared	Enumeration Literal	AggregationKind	<code>\$AggregationKind[i].shared</code>

47. Alias

A UPDM Artifact used to define an alternative name for an element as used by DoDAF or MODAF.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Alias[i].actualPropertySet</code>
Allias.annotatedElement		Alias	<code>\$Alias[i].Allias.annotatedElement</code>
conformsTo	Standard	UPDMElement	<code>\$Alias[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Alias[i].endBoundaryType</code>
nameOwner	String	Alias	<code>\$Alias[i].nameOwner</code>
propertySet	PropertySet	UPDMElement	<code>\$Alias[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Alias[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Alias[i].URI</code>

48. All Views Report**Base Classifier**

- [InvisibleStereotype](#)

49. All Views Viewpoint**Base Classifier**

- [InvisibleStereotype](#)

50. Allocate

Allocate is a dependency based on UML::abstraction. It is a mechanism for associating elements of different types, or in different hierarchies, at an abstract level. Allocate is used for assessing user model consistency and directing future design activity. It is expected that an «allocate» relationship between model elements is a precursor to a more concrete relationship between the elements, their properties, operations, attributes, or sub-classes.

Base Classifier

- [DirectedRelationshipPropertyPath](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getAllocatedFrom	NamedElement	Allocate	<code>\$Allocate[i].getAllocatedFrom</code>
getAllocatedTo	NamedElement	Allocate	<code>\$Allocate[i].getAllocatedTo</code>
sourceContext	Classifier	DirectedRelationshipPropertyPath	<code>\$Allocate[i].sourceContext</code>
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$Allocate[i].sourcePropertyPath</code>
targetContext	Classifier	DirectedRelationshipPropertyPath	<code>\$Allocate[i].targetContext</code>
targetPropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$Allocate[i].targetPropertyPath</code>

51. AllocateActivityPartition

AllocateActivityPartition is used to depict an «allocate» relationship on an Activity diagram. The AllocateActivityPartition is a standard UML2::ActivityPartition, with modified constraints as stated in the paragraph below.

52. Allocated

«allocated» is a stereotype that applies to any NamedElement that has at least one allocation relationship with another NamedElement. «allocated» elements may be designated by either the /from or /to end of an «allocate» dependency. The «allocated» stereotype provides a mechanism for a particular model element to conveniently retain and display the element at the opposite end of any «allocate» dependency. This stereotype provides for the properties “allocatedFrom” and “allocatedTo,” which are derived from the «allocate» dependency.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
allocatedFrom	NamedElement	Allocated	<code>\$Allocated[i].allocatedFrom</code>
allocatedTo	NamedElement	Allocated	<code>\$Allocated[i].allocatedTo</code>

53. ApprovalStatus

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
approved	Enumeration Literal	ApprovalStatus	<code>\$ApprovalStatus[i].approved</code>
not approved	Enumeration Literal	ApprovalStatus	<code>\$ApprovalStatus[i].not approved</code>

54. ArbitraryConnector

UPDM: Represents a visual indication of a connection used in high level operational concept diagrams. The connections are purely visual and cannot be related to any architectural semantics.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ArbitraryConnector[i].actualPropertySet</code>
ArbitraryRelationship.client		ArbitraryConnector	<code>\$ArbitraryConnector[i].ArbitraryRelationship.client</code>
ArbitraryRelationship.supplier		ArbitraryConnector	<code>\$ArbitraryConnector[i].ArbitraryRelationship.supplier</code>
conformsTo	Standard	UPDMElement	<code>\$ArbitraryConnector[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ArbitraryConnector[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ArbitraryConnector[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ArbitraryConnector[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ArbitraryConnector[i].URI</code>

55. ArbitraryRelationshipDashedLeft

Base Classifier

- [InvisibleStereotype](#)

56. ArbitraryRelationshipDashedRight

Base Classifier

- [InvisibleStereotype](#)

57. ArbitraryRelationshipDirectedLeft

Base Classifier

- [InvisibleStereotype](#)

58. ArbitraryRelationshipDirectedRight

Base Classifier

- [InvisibleStereotype](#)

59. ArbitraryRelationshipSolid

Base Classifier

- [InvisibleStereotype](#)

60. ArchitecturalDescription

MODAF: A specification of a system of systems at a technical level which also provides the business context for the system of systems.

DoDAF: Information describing an architecture such as an OV-5 Activity Model document.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ArchitecturalDescription[i].actualPropertySet</code>
approvalAuthority	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].approvalAuthority</code>
architect	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].architect</code>
ArchitecturalDescription.architectureFramework		ArchitecturalDescription	<code>\$ArchitecturalDescription[i].ArchitecturalDescription.architectureFramework</code>
architectureFramework	ArchitectureFrameworkKind	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].architectureFramework</code>
assumptionAndConstraint	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].assumptionAndConstraint</code>
conformsTo	Standard	UPDMElement	<code>\$ArchitecturalDescription[i].conformsTo</code>

creatingOrganization	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].creatingOrganization</code>
dateCompleted	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].dateCompleted</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ArchitecturalDescription[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ArchitecturalDescription[i].propertySet</code>
purpose	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].purpose</code>
recommendations	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].recommendations</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ArchitecturalDescription[i].startBoundaryType</code>
summaryOfFindings	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].summaryOfFindings</code>
toBe	Boolean	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].toBe</code>
toolsUsed	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].toolsUsed</code>
URI	String	UPDMElement	<code>\$ArchitecturalDescription[i].URI</code>
viewpoint	String	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].viewpoint</code>
views	View	ArchitecturalDescription	<code>\$ArchitecturalDescription[i].views</code>

61. ArchitecturalReference

MODAF: Asserts that one architectural description (referrer) refers to another (referred).

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ArchitecturalReference[i].actualPropertySet</code>
ArchitecturalReference.client		ArchitecturalReference	<code>\$ArchitecturalReference[i].ArchitecturalReference.client</code>
ArchitecturalReference.supplier		ArchitecturalReference	<code>\$ArchitecturalReference[i].ArchitecturalReference.supplier</code>
conformsTo	Standard	UPDMElement	<code>\$ArchitecturalReference[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ArchitecturalReference[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ArchitecturalReference[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ArchitecturalReference[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ArchitecturalReference[i].URI</code>

62. Architecture Meta-Data Viewpoint

Base Classifier

- [InvisibleStereotype](#)

63. ArchitectureFrameworkKind

Architecture Framework Kind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
DoDAF	Enumeration Literal	ArchitectureFrameworkKind	<code>\$ArchitectureFrameworkKind[i].DoDAF</code>
DoDAF 2.0	Enumeration Literal	ArchitectureFrameworkKind	<code>\$ArchitectureFrameworkKind[i].DoDAF 2.0</code>
MODAF	Enumeration Literal	ArchitectureFrameworkKind	<code>\$ArchitectureFrameworkKind[i].MODAF</code>
NAF	Enumeration Literal	ArchitectureFrameworkKind	<code>\$ArchitectureFrameworkKind[i].NAF</code>
NAF 4.0	Enumeration Literal	ArchitectureFrameworkKind	<code>\$ArchitectureFrameworkKind[i].NAF 4.0</code>

64. ArchitectureIntroduction**Base Classifier**

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
architectureFramework	ArchitectureFrameworkKind	ArchitectureIntroduction	<code>\$ArchitectureIntroduction[i].architectureFramework</code>

65. ArchitectureMetadata

UPDM: Information on ArchitecturalDescription. It states things like what methodology was used, notation, etc.

MODAF: A Metadata element that applies to the whole architecture.

DoDAF: NA

Base Classifier

- [Metadata](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ArchitectureMetadata[i].actualPropertySet</code>
ArchitectureMetadata.annotatedElement		ArchitectureMetadata	<code>\$ArchitectureMetadata[i].ArchitectureMetadata.annotatedElement</code>
conformsTo	Standard	UPDMElement	<code>\$ArchitectureMetadata[i].conformsTo</code>
dublinCoreElement	String	Metadata	<code>\$ArchitectureMetadata[i].dublinCoreElement</code>

endBoundaryType	ISO8601DateTime	UPDMElement	\$ArchitectureMetadata[i].endBoundaryType
modMetaDataElement	String	Metadata	\$ArchitectureMetadata[i].modMetaDataElement
name	String	Metadata	\$ArchitectureMetadata[i].name
propertySet	PropertySet	UPDMElement	\$ArchitectureMetadata[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ArchitectureMetadata[i].startBoundaryType
URI	String	UPDMElement	\$ArchitectureMetadata[i].URI

66. assembly

67. AssociationOfInformation

MODAF: Asserts that there is a relationship between two entities (Entity Relationship).

DoDAF: A relationship or association between two elements of information.

Base Classifier

- [EntityRelationship](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$AssociationOfInformation[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$AssociationOfInformation[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$AssociationOfInformation[i].endBoundaryType
EntityRelationship.endType		EntityRelationship	\$AssociationOfInformation[i].EntityRelationship.endType
propertySet	PropertySet	UPDMElement	\$AssociationOfInformation[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$AssociationOfInformation[i].startBoundaryType
URI	String	UPDMElement	\$AssociationOfInformation[i].URI

68. AsynchronousMessage

MODAF: A signal which is transmitted irregularly with respect to time.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$AsynchronousMessage[i].actualPropertySet

conformsTo	Standard	UPDMElement	\$AsynchronousMessage[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$AsynchronousMessage[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$AsynchronousMessage[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$AsynchronousMessage[i].startBoundaryType
URI	String	UPDMElement	\$AsynchronousMessage[i].URI

69. AttachedFile

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
attachedAt	date	AttachedFile	\$AttachedFile[i].attachedAt
author	String	AttachedFile	\$AttachedFile[i].author
file	String	AttachedFile	\$AttachedFile[i].file
modifiedAt	date	AttachedFile	\$AttachedFile[i].modifiedAt
size	String	AttachedFile	\$AttachedFile[i].size

70. Attachment

A part of a Message that is attached to rather than contained in the message.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
encoding	String	Attachment	\$Attachment[i].encoding
mimeType	String	Attachment	\$Attachment[i].mimeType

71. autoGeneratedName

Base Classifier

- InvisibleStereotype

72. AutoImageSize

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Fit and rotate image (clockwise) to paper (large only)	Enumeration Literal	AutoImageSize	\$AutoImageSize[i].Fit and rotate image (clockwise) to paper (large only)

Fit and rotate image (counter-clockwise) to paper (large only)	Enumeration Literal	AutoImageSize	<code>\$AutoImageSize[i].Fit and rotate image (counter-clockwise) to paper (large only)</code>
Fit image to paper (large only)	Enumeration Literal	AutoImageSize	<code>\$AutoImageSize[i].Fit image to paper (large only)</code>
No Resize	Enumeration Literal	AutoImageSize	<code>\$AutoImageSize[i].No Resize</code>

73. AutoNumber

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
defaultNumber	Boolean	AutoNumber	<code>\$AutoNumber[i].defaultNumber</code>
numberedProperty	Property	AutoNumber	<code>\$AutoNumber[i].numberedProperty</code>
numberingScheme	NumberingScheme	AutoNumber	<code>\$AutoNumber[i].numberingScheme</code>
prefix	String	AutoNumber	<code>\$AutoNumber[i].prefix</code>
usePrefixOnOwner	boolean	AutoNumber	<code>\$AutoNumber[i].usePrefixOnOwner</code>

74. Auxiliary

A class that supports another more central or fundamental class, typically by implementing secondary logic or control flow. The class that the auxiliary supports may be defined explicitly using a Focus class or implicitly by a dependency relationship. Auxiliary classes are typically used together with Focus classes, and are particularly useful for specifying the secondary business logic or control flow of components during design. See also: «focus».

75. auxiliaryResource

Base Classifier

- [InvisibleStereotype](#)

76. AV-1

Base Classifier

- [InvisibleStereotype](#)

77. AV-1 DARS Report

Base Classifier

- [InvisibleStereotype](#)

78. AV-1 Package

Base Classifier

- [InvisibleStereotype](#)

79. AV-1 Report

Base Classifier

- [InvisibleStereotype](#)

80. AV-2

The Integrated Dictionary (AV-2) provides definitions of all terms used throughout the architectural data.

To fill in AV-2 table, UPDM elements have to be added to it.

*Click **Add Element** button to create a new or to add an existing element to the table.*

"Name", "Definition", "Alias", "Same As", "Documentation", and wide range of column cells are allowed to edit in the table. "UPDM Type", "UML Metatype", "SysML Type", "BPMN Type" cells are read only.

Rows (UPDM Elements) can be removed from the model or only from the table, can be ordered, and exported to the CSV or HTML. Four kind of reports can be printed reflecting the data shown in the table.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	AV-2	\$AV-2[i].hideColumns

81. AV-2 Package

Base Classifier

- [InvisibleStereotype](#)

82. AV-2 Report

Base Classifier

- [InvisibleStereotype](#)

83. AV-2 Tabular Report

Base Classifier

- [InvisibleStereotype](#)

84. BasicInterval

Basic Interval distribution - value between min and max inclusive

Base Classifier

- [DistributedProperty](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
max	Real	BasicInterval	\$BasicInterval[i].max
min	Real	BasicInterval	\$BasicInterval[i].min

85. BindingConnector

A *Binding Connector* is a connector which specifies that the properties at both ends of the connector have equal values. If the properties at the ends of a binding connector are typed by a *DataType* or *ValueType*, the connector specifies that the instances of the properties must hold equal values, recursively through any nested properties within the connected properties. If the properties at the ends of a binding connector are typed by a *Block*, the connector specifies that the instances of the properties must refer to the same block instance. As with any connector owned by a *SysML Block*, the ends of a binding connector may be nested within a multi-level path of properties accessible from the owning block. The *NestedConnectorEnd* stereotype is used to represent such nested ends just as for nested ends of other *SysML* connectors.



86. BlackBoxICDTable

A *Blackbox ICD Table* represents all external Ports and interfaces of the *Block*.

With this table you can easily:

- Review interfaces (Ports) of the *Block* in the single place.
- Customize the representation of the table.
- Export the data into an *.html, *.csv, or *.xlsx file.

Toolbar button descriptions:

- **Delete** – click to remove selected elements both from the table and from the model.
- **Remove From Table** - click to remove selected elements from the table.
- **Refresh** – click  to update the contents of the table after specifying the *Block*.
- **Validate Diagram** – click  to validate the diagram.
- **Up** – click to shift selected elements (either grouped or non-grouped) up a row.
- **Down** – click to shift selected elements (either grouped or non-grouped) down a row.
- **Show Columns** – click to specify the set of columns for displaying in the table.
- **Options > Show Full Paths** – click to display the full paths of elements in the table.
- **Export** - click to export the contents of the table to an *.html, *.csv, or *.xlsx file.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
query	Element	BlackBoxICDTable	\$BlackBoxICDTable[i].query

87. Block

A *Block* is a modular unit that describes the structure of a system or element. It may include both structural and behavioral features, such as properties and operations, that represent the state of the system and behavior that the system may exhibit. Some of these properties may hold parts of a system, which can also be described by blocks. A block may include a structure of connectors between its properties to indicate how its parts or other properties relate to one another. *SysML* blocks provide a general-

purpose capability to describe the architecture of a system. They provide the ability to represent a system hierarchy, in which a system at one level is composed of systems at a more basic level. They can describe not only the connectivity relationships between the systems at any level, but also quantitative values or other information about a system. SysML does not restrict the kind of system or system element that may be described by a block. Any reusable form of description that may be applied to a system or a set of system characteristics may be described by a block. Such reusable descriptions, for example, may be applied to purely conceptual aspects of a system design, such as relationships that hold between parts or properties of a system. Connectors owned by SysML blocks may be used to define relationships between parts or other properties of the same containing block. The type of a connector or its connected ends may specify the semantic interpretation of a specific connector.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	\$Block[i].isEncapsulated

88. BlockHierarchy

Block definition diagram usage for a block hierarchy - Block Hierarchy where block can be replaced by system, item, activity, etc.

89. BlockProperty

Obsolete.

Base Classifier

- [InvisibleStereotype](#)

90. Boolean

91. boundary

A boundary is a class that lies on the periphery of a system, but within it. It interacts with actors outside the system as well as with entity, control, and other boundary classes within the system.

92. Boundary system

A Boundary system is a special external system that serves as medium between another system and the system under development without having own interests in the communication. For example Bus system or Communication system.

Base Classifier

- [External system](#)

93. BoundReference

Base Classifier

- [EndPathMultiplicity](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
bindingPath	Property	BoundReference	<code>\$BoundReference[i].bindingPath</code>
boundEnd	ConnectorEnd	BoundReference	<code>\$BoundReference[i].boundEnd</code>
lower	Integer	EndPathMultiplicity	<code>\$BoundReference[i].lower</code>
upper	UnlimitedNatural	EndPathMultiplicity	<code>\$BoundReference[i].upper</code>

94. BuildComponent

A collection of elements defined for the purpose of system level development activities, such as compilation and versioning.

95. businessRequirement

High-level business requirement.

Base Classifier

- [extendedRequirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	<code>\$businessRequirement[i].Derived</code>
DerivedFrom	Requirement	Requirement	<code>\$businessRequirement[i].DerivedFrom</code>
Id	String	Requirement	<code>\$businessRequirement[i].Id</code>
Master	Requirement	Requirement	<code>\$businessRequirement[i].Master</code>
RefinedBy	NamedElement	Requirement	<code>\$businessRequirement[i].RefinedBy</code>
risk	RiskKind	extendedRequirement	<code>\$businessRequirement[i].risk</code>
SatisfiedBy	NamedElement	Requirement	<code>\$businessRequirement[i].SatisfiedBy</code>
source	String	extendedRequirement	<code>\$businessRequirement[i].source</code>
Text	String	Requirement	<code>\$businessRequirement[i].Text</code>
TracedTo	NamedElement	Requirement	<code>\$businessRequirement[i].TracedTo</code>
VerifiedBy	NamedElement	Requirement	<code>\$businessRequirement[i].VerifiedBy</code>
verifyMethod	VerificationMethodKind	extendedRequirement	<code>\$businessRequirement[i].verifyMethod</code>

96. C1

Base Classifier

- [InvisibleStereotype](#)

97. C1 Package**Base Classifier**

- [InvisibleStereotype](#)

98. C1-S1

The Capability to Services Mapping Matrix (C1-S1) depicts which Service Interfaces contribute to the achievement of a Capability. The Rows of this matrix are Service Interfaces and the Columns are Capabilities.

To build the Matrix:

- 1. Specify Rows scope (Service Interfaces);*
- 2. Specify Columns scope (Capabilities);*
- 3. Click "Refresh" button.*

Service Interfaces expose Capabilities using "Expose" relationship.

To map Service Interface to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

99. C2**Base Classifier**

- [InvisibleStereotype](#)

100. C2 Package

Base Classifier

- [InvisibleStereotype](#)

101. C3

Base Classifier

- [InvisibleStereotype](#)

102. C3 Package

Base Classifier

- [InvisibleStereotype](#)

103. C4

The Standard Processes (C4) describes the mapping between the capabilities required by an Enterprise and the operational activities that those capabilities support. The Rows of this matrix are Capabilities and the Columns are Standard Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Capabilities);*
- 2. Specify Columns scope (Standard Operational Activities);*
- 3. Click "Refresh" button.*

Standard Operational Activities maps to Capabilities using "Maps to Capability" relationship.

To map Standard Operational Activity to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

104. C4 Package

Base Classifier

- [InvisibleStereotype](#)

105. C5

Base Classifier

- [InvisibleStereotype](#)

106. C5 Package

Base Classifier

- [InvisibleStereotype](#)

107. C7 Actual Performance Parameters

Actual Performance Parameters (C7 Actual) depicts the Actual values of performance characteristics of a Capability. There are three ways to add a row in this table:

1. **Add new** measurable Capability; Click **Add New** button and select one or more Capabilities that have at least one Measurement Set Defined (see C7 Typical). Specify values for each Measurement - directly in the table cells.
2. **Add existing** Measures or measurable Capabilities. Click **Add Existing** button and select Existing Measurements or Capabilities.
3. **Add missing** Measurements. Click “**Add the missing Measurements**” button to update table to include latest model changes.

Rows (Actual Measurements) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

hideColumns	String	C7 Actual Performance Parameters	\$C7ActualPerformanceParameters[i].hideColumns
-------------	--------	--	--

108. C7 Package

Base Classifier

- [InvisibleStereotype](#)

109. C7 Performance Parameters

Performance Parameters (C7) depicts the possible types of performance characteristics of a Capability. There are two ways to add a row in this table:

1. *Add new Measurement Set. Click **Add New** button and select the owning element for Measurement Set. Specify Measurements to the Measurement Set and Capabilities to be Measured - straight in the table cells.*
2. *Add existing Measurements Sets. Click **Add Existing** button and select Existing Measurement Sets.*

Rows (Measurement Sets) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	C7 Performance Parameters	\$C7PerformanceParameters[i].hideColumns

110. C7 Report

Base Classifier

- [InvisibleStereotype](#)

111. Call

A usage dependency whose source is an operation and whose target is an operation. The relationship may also be subsumed to the class containing an operation, with the meaning that there exists an operation in the class to which

the dependency applies.

A call dependency specifies that the source operation or an operation in the source class invokes the target operation or an operation in the target class. A call dependency may connect a source operation to any target operation that is within scope including, but not limited to, operations of the enclosing classifier and operations of other visible classifiers.

112. CallConcurrencyKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
concurrent	Enumeration Literal	CallConcurrencyKind	<code>\$CallConcurrencyKind[i].concurrent</code>
guarded	Enumeration Literal	CallConcurrencyKind	<code>\$CallConcurrencyKind[i].guarded</code>
sequential	Enumeration Literal	CallConcurrencyKind	<code>\$CallConcurrencyKind[i].sequential</code>

113. Capability

MODAF: A high level specification of the enterprise's ability.

DoDAF: The ability to achieve a desired effect under specified [performance] standards and conditions through combinations of ways and means [activities and resources] to perform a set of activities.

Base Classifier

- [Capability](#)
- [Desirer](#)
- [PropertySet](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Capability[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Capability[i].appliesTo</code>
Capability.ownedAttribute		Capability	<code>\$Capability[i].Capability.ownedAttribute</code>
conformsTo	Standard	UPDMElement	<code>\$Capability[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Capability[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Capability[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Capability[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Capability[i].URI</code>

114. Capability

A Capability is the ability to act and produce an outcome that achieves a result. It can specify a general capability of a participant as well as the specific ability to provide a service.

115. Capability Behavioral Map

*A **Capability Behavioral Map** depicts the behavioral relationships of a Capability. The predefined map includes: the **Capability** itself, related **Operational Activities**, and related **System Functions**. A Capability Behavioral Map also presents the internal decomposition of each mapped behavioral element as well as the relationships among these elements.*

*The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:*



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:


***Create a relation map structure** - drag and drop element from the Containment Tree on the Relation Map.*

***Restore manually suppressed / expanded branches and hidden elements and they position** - click Restore Layout button .*

***Expand / suppress branches** - click on smart manipulator after the Node /.*

***Move the whole structure** - click on the empty place in the Relation Map and drag.*

***Move the selected Node** - click on the Node and drag.*

***Insert New Element** - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.*

***Insert New Element of the Same Type** - type the name and press Ctrl+Enter to create an element of the same type.*

***Zoom in** - Ctrl + mouse wheel scroll up.*

***Zoom out** - Ctrl + mouse wheel scroll down.*

***Change the Context element on demand by selecting a new Node** – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.*

***Change the Context element** – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .*

Base Classifier

- InvisibleStereotype

116. Capability Structural Map

A **Capability Structural Map** depicts the structural relationships of a Capability. The predefined map includes: the **Capability** itself, related **Performers** (MODAF Nodes), and related **System Resources** including Capability Configurations, Organization Types, Software and other elements. A Capability Structural Map also presents the internal compositions of each mapped element as well as the relationships among these elements.

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:


Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node  / .

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Insert New Element - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.

Insert New Element of the Same Type - type the name and press Ctrl+Enter to create an element of the same type.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

117. Capability Viewpoint

Base Classifier

- [InvisibleStereotype](#)

118. CapabilityConfiguration

MODAF: A composite structure representing the physical and human resources (and their interactions) in an enterprise.--A CapabilityConfiguration is a set of artefacts or an organisation configured to provide a capability, and should be guided by [doctrine] which may take the form of Standard or OperationalConstraint stereotypes.

DoDAF: Any entity - human, automated, or any aggregation of human and/or automated - that performs an activity and provides a capability (Performer).

Base Classifier

- [PhysicalArchitecture](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$CapabilityConfiguration[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$CapabilityConfiguration[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$CapabilityConfiguration[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$CapabilityConfiguration[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$CapabilityConfiguration[i].conformsTo</code>
doctrine	Constraint	CapabilityConfiguration	<code>\$CapabilityConfiguration[i].doctrine</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityConfiguration[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$CapabilityConfiguration[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$CapabilityConfiguration[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$CapabilityConfiguration[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$CapabilityConfiguration[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$CapabilityConfiguration[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$CapabilityConfiguration[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$CapabilityConfiguration[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityConfiguration[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$CapabilityConfiguration[i].URI</code>

119. CapabilityConfigurationConceptRole

Base Classifier

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$CapabilityConfigurationConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$CapabilityConfigurationConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$CapabilityConfigurationConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityConfigurationConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$CapabilityConfigurationConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityConfigurationConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$CapabilityConfigurationConceptRole[i].URI</code>

120. CapabilityOfPerformer

UPDM: A couple that represents the capability that a resource, node or enterprise phase exhibits (Exhibits).

MODAF: An assertion that a Node is required to have a Capability (Capability for node).

DoDAF: A couple that represents the capability that a performer has.

Base Classifier

- [Exhibits](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$CapabilityOfPerformer[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$CapabilityOfPerformer[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityOfPerformer[i].endBoundaryType</code>
environmentalConditions	Environment	Exhibits	<code>\$CapabilityOfPerformer[i].environmentalConditions</code>
Exhibits.client		Exhibits	<code>\$CapabilityOfPerformer[i].Exhibits.client</code>
Exhibits.supplier		Exhibits	<code>\$CapabilityOfPerformer[i].Exhibits.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$CapabilityOfPerformer[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityOfPerformer[i].startBoundaryType</code>
universalCapabilitySet	ActualPropertySet	Exhibits	<code>\$CapabilityOfPerformer[i].universalCapabilitySet</code>
URI	String	UPDMElement	<code>\$CapabilityOfPerformer[i].URI</code>

121. CapabilityProperty

UPDM: A property of a capability.

MODAF: NA

DoDAF: NA

Base Classifier

- Property

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$CapabilityProperty[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$CapabilityProperty[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityProperty[i].endBoundaryType</code>
maxValue	String	Property	<code>\$CapabilityProperty[i].maxValue</code>
minValue	String	Property	<code>\$CapabilityProperty[i].minValue</code>
propertySet	PropertySet	UPDMElement	<code>\$CapabilityProperty[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapabilityProperty[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$CapabilityProperty[i].URI</code>

122. CapableElement

UPDM An abstract element that represents a structural element that can perform behaviors (i.e. *PerformedActivity*).

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$CapableElement[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$CapableElement[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapableElement[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$CapableElement[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CapableElement[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$CapableElement[i].URI</code>

123. ChangeStructuralFeatureEvent

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
structuralFeature	StructuralFeature	ChangeStructuralFeatureEvent	<code>\$ChangeStructuralFeatureEvent[i].structuralFeature</code>

124. ClassificationType

Enumeration of types of security classification, derived from DoDAF.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
C	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].C</code>
CTS	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].CTS</code>
CTS-B	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].CTS-B</code>
CTS-BALK	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].CTS-BALK</code>
CTSA	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].CTSA</code>
NC	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NC</code>
NCA	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NCA</code>
NR	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NR</code>
NS	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NS</code>
NS-A	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NS-A</code>
NS-S	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NS-S</code>
NSAT	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NSAT</code>
NU	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].NU</code>
R	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].R</code>
S	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].S</code>
TS	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].TS</code>
U	Enumeration Literal	ClassificationType	<code>\$ClassificationType[i].U</code>

125. ClassifierBehaviorProperty

The ClassifierBehaviorProperty stereotype can be applied to properties to constrain their values to be the executions of classifier behaviors. The value of properties with ClassifierBehaviorProperty applied are the executions of classifier behaviors invoked by instantiation of the block that owns the stereotyped property or one of its specializations.

126. Climate

MODAF: A type of weather condition, or combination of weather conditions (e.g. high temperature & dry).

DoDAF: NA

Base Classifier

- [Environment](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Climate[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Climate[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Climate[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Climate[i].endBoundaryType</code>
Environment.ownedAttributes		Environment	<code>\$Climate[i].Environment.ownedAttributes</code>
propertySet	PropertySet	UPDMElement	<code>\$Climate[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Climate[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Climate[i].URI</code>

127. Collaboration

Abstract stereotype for Service Contract and Service Architecture common features

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isStrict	Boolean	Collaboration	<code>\$Collaboration[i].isStrict</code>

128. colorHolder

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
alpha	Integer	colorHolder	<code>\$colorHolder[i].alpha</code>
b	Integer	colorHolder	<code>\$colorHolder[i].b</code>
g	Integer	colorHolder	<code>\$colorHolder[i].g</code>
r	Integer	colorHolder	<code>\$colorHolder[i].r</code>

129. Command

MODAF: Asserts that one OrganisationalResource (source) commands another (target)

DoDAF: NA

Base Classifier

- [ResourceInteraction](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Command[i].actualPropertySet</code>
Command.conveyed		Command	<code>\$Command[i].Command.conveyed</code>
Command.informationSource		Command	<code>\$Command[i].Command.informationSource</code>
Command.informationTarget		Command	<code>\$Command[i].Command.informationTarget</code>
conformsTo	Standard	UPDMElement	<code>\$Command[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Command[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Command[i].propertySet</code>
ResourceInteraction.conveyedElement		ResourceInteraction	<code>\$Command[i].ResourceInteraction.conveyedElement</code>
ResourceInteraction.informationSource		ResourceInteraction	<code>\$Command[i].ResourceInteraction.informationSource</code>
ResourceInteraction.informationTarget		ResourceInteraction	<code>\$Command[i].ResourceInteraction.informationTarget</code>
ResourceInteraction.realization		ResourceInteraction	<code>\$Command[i].ResourceInteraction.realization</code>
ResourceInteraction.realizingActivityEdge		ResourceInteraction	<code>\$Command[i].ResourceInteraction.realizingActivityEdge</code>
ResourceInteraction.realizingConnector		ResourceInteraction	<code>\$Command[i].ResourceInteraction.realizingConnector</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Command[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Command[i].URI</code>

130. Competence

MODAF: A specific set of abilities defined by knowledge, skills and attitude.

DoDAF: (DoDAF::Skill): The ability, coming from one's knowledge, practice, aptitude, etc., to do something well.

Base Classifier

- [PropertySet](#)
- [SubjectOfForecast](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Competence[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Competence[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Competence[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Competence[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Competence[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Competence[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Competence[i].URI</code>

131. CompetenceProvider

UPDM: Abstract element used to group ActualPersons and ActualOrganisationalResources.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$CompetenceProvider[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$CompetenceProvider[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CompetenceProvider[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$CompetenceProvider[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CompetenceProvider[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$CompetenceProvider[i].URI</code>

132. CompetenceRequirer

UPDM: Abstract element used to group Organizations, Post and Responsibilities.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$CompetenceRequirer[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$CompetenceRequirer[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CompetenceRequirer[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$CompetenceRequirer[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$CompetenceRequirer[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$CompetenceRequirer[i].URI</code>

133. CompletionStatus

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
complete	Enumeration Literal	CompletionStatus	<code>\$CompletionStatus[i].complete</code>

draft	Enumeration Literal	CompletionStatus	<code>\$CompletionStatus[i].draft</code>
under analysis	Enumeration Literal	CompletionStatus	<code>\$CompletionStatus[i].under analysis</code>
under development	Enumeration Literal	CompletionStatus	<code>\$CompletionStatus[i].under development</code>

134. Complex

A *Complex value type* represents the mathematical concept of a complex number. A complex number consists of a real part defined by a real number, and an imaginary part defined by a real number multiplied by the square root of -1. Complex numbers are used to express solutions to various forms of mathematical equations.

Base Classifier

- [Number](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
imaginaryPart	Real	Complex	<code>\$Complex[i].imaginaryPart</code>
realPart	Real	Complex	<code>\$Complex[i].realPart</code>

135. Composite

Base Classifier

- [Participant](#)

136. ConceptItem

UPDM: Abstract, an item which may feature in a high level operational concept.

DoDAF:NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ConceptItem[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ConceptItem[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConceptItem[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ConceptItem[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConceptItem[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ConceptItem[i].URI</code>

137. ConceptRole

UPDM: A relationship which asserts that a ConceptItem forms part of the high level operational concept.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$ConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$ConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ConceptRole[i].URI</code>

138. Concepts Viewpoint

Base Classifier

- [InvisibleStereotype](#)

139. conceptualView

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
conceptualViewID	int	conceptualView	<code>\$conceptualView[i].conceptualViewID</code>

140. Condition

MODAF: A definition of the conditions in which something exists or functions. An Environment may be specified in terms of LocationType (e.g. terrain), Climate (e.g. tropical), and LightCondition (e.g. dark, light, dusk, etc.)

DoDAF: An object that encompasses meteorological, geographic, and control features mission significance.

Base Classifier

- Environment

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Condition[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Condition[i].appliesTo</code>
Condition.ownedAttribute		Condition	<code>\$Condition[i].Condition.ownedAttribute</code>
conditionKind	String	Condition	<code>\$Condition[i].conditionKind</code>
conformsTo	Standard	UPDMElement	<code>\$Condition[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Condition[i].endBoundaryType</code>
Environment.ownedAttributes		Environment	<code>\$Condition[i].Environment.ownedAttributes</code>
propertySet	PropertySet	UPDMElement	<code>\$Condition[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Condition[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Condition[i].URI</code>

141. ConditionProperty

MODAF: EnvironmentalProperty: Asserts that an Environment has one or more properties. These may be Climate, LocationType, or LightCondition.

DoDAF: NA

Base Classifier

- [EnvironmentProperty](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ConditionProperty[i].actualPropertySet</code>
ConditionProperty.class		ConditionProperty	<code>\$ConditionProperty[i].ConditionProperty.class</code>
ConditionProperty.type		ConditionProperty	<code>\$ConditionProperty[i].ConditionProperty.type</code>
conformsTo	Standard	UPDMElement	<code>\$ConditionProperty[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConditionProperty[i].endBoundaryType</code>
EnvironmentalProperty.class		EnvironmentProperty	<code>\$ConditionProperty[i].EnvironmentalProperty.class</code>
EnvironmentalProperty.type		EnvironmentProperty	<code>\$ConditionProperty[i].EnvironmentalProperty.type</code>
maxValue	String	Property	<code>\$ConditionProperty[i].maxValue</code>
minValue	String	Property	<code>\$ConditionProperty[i].minValue</code>
propertySet	PropertySet	UPDMElement	<code>\$ConditionProperty[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConditionProperty[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ConditionProperty[i].URI</code>

142. ConditionType

Abstract element indicating what an EnvironmentProperty can be typed by.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ConditionType[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ConditionType[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConditionType[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ConditionType[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ConditionType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ConditionType[i].URI</code>

143. Conform

A Conform relationship is a dependency between a view and a viewpoint. The view conforms to the specified rules and conventions detailed in the viewpoint. Conform is a specialization of the UML dependency, and as with other dependencies the arrow direction points from the (client/source) to the (supplier/target).

144. ConnectorKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
assembly	Enumeration Literal	ConnectorKind	<code>\$ConnectorKind[i].assembly</code>
delegation	Enumeration Literal	ConnectorKind	<code>\$ConnectorKind[i].delegation</code>

145. ConnectorProperty

Connectors can be typed by association classes that are stereotyped by Block (association blocks). These connectors specify instances (links) of the association block that exist due to instantiation of the block owning or inheriting the connector. The value of a connector property on an instance of a block will be exactly those link objects that are instances of the association block typing the connector referred to by the connector property.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
connector	Connector	ConnectorProperty	<code>\$ConnectorProperty[i].connector</code>

146. ConstraintBlock

A constraint block is a block that packages the statement of a constraint so it may be applied in a reusable way to constrain properties of other blocks. A constraint block typically defines one or more constraint parameters, which are bound to properties of other blocks in a surrounding context where the constraint is used. Binding connectors, as defined in Chapter 8: Blocks, are used to bind each parameter of the constraint block to a property in the surrounding context. All properties of a constraint block are constraint parameters, with the exception of constraint properties that hold internally nested usages of other constraint blocks.

Base Classifier

- [Block](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	\$ConstraintBlock[i].isEncapsulated

147. ConstraintParameter

A parametric diagram may contain constraint properties and their parameters, along with other properties from within the internal block context. All properties that appear, other than the constraints themselves, must either be bound directly to a constraint parameter, or contain a property that is bound to one (through any number of levels of containment).

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
additional_stereotypes		ConstraintParameter	\$ConstraintParameter[i].additional_stereotypes
ConstraintParameter		ConstraintParameter	\$ConstraintParameter[i].ConstraintParameter

148. ConstraintProperty

A constraint property is a property of any block that is typed by a constraint block. It holds a localized usage of the constraint block. Binding connectors may be used to bind the parameters of this constraint block to other properties of the block that contains the usage.

Base Classifier

- [InvisibleStereotype](#)

149. constructor

150. Consumer

Defines Interface for consumer role in Service Contact.

151. Content

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
elementTypes	Element	Content	<code>\$Content[i].elementTypes</code>
excludedElementTypes	Element	Content	<code>\$Content[i].excludedElementTypes</code>

152. ContextDiagram

A user defined usage of an internal block diagram, which depicts some of the top level entities in the overall enterprise and their relationships.

153. contextSpecificDefaultValue

Obsolete.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
path	Property	contextSpecificDefaultValue	<code>\$contextSpecificDefaultValue[i].path</code>

154. contextSpecificValuesHolder

Obsolete.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
instances	InstanceSpecification	contextSpecificValuesHolder	<code>\$contextSpecificValuesHolder[i].instances</code>

155. Continuous

Continuous rate is a special case of rate of flow (see Rate) where the increment of time between items approaches zero. It is intended to represent continuous flows that may correspond to water flowing through a pipe, a time continuous signal, or continuous energy flow. It is independent from UML streaming. A streaming parameter may or may not apply to continuous flow, and a continuous flow may or may not apply to streaming parameters.

Base Classifier

- [Rate](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
rate	InstanceSpecification	Rate	\$Continuous[i].rate

156. Contract

A specialization of an “OperationalExchange” a “Contract” specifies an agreement between two or more parties to exchange information. The Contract forms an ontological commitment between parties in a community of interest (CoI) or Community of Practice (CoP). The contract is also used to realize the information exchange requirements of either a needline or a community of interest.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Contract.conveyed		Contract	\$Contract[i].Contract.conveyed

157. Control

MODAF: A type of ResourceInteraction where one Resource (source) controls another (target). --Examples - the driver of a tank, one organisation having operational control of another, a fire control system controlling a weapons system.

DoDAF: NA

Base Classifier

- [ResourceInteraction](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Control[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Control[i].conformsTo
Control.conveyed		Control	\$Control[i].Control.conveyed
Control.informationSource		Control	\$Control[i].Control.informationSource
Control.informationTarget		Control	\$Control[i].Control.informationTarget
endBoundaryType	ISO8601DateTime	UPDMElement	\$Control[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$Control[i].propertySet
ResourceInteraction.conveyedElement		ResourceInteraction	\$Control[i].ResourceInteraction.conveyedElement
ResourceInteraction.informationSource		ResourceInteraction	\$Control[i].ResourceInteraction.informationSource
ResourceInteraction.informationTarget		ResourceInteraction	\$Control[i].ResourceInteraction.informationTarget
ResourceInteraction.realization		ResourceInteraction	\$Control[i].ResourceInteraction.realization

ResourceInteraction.realizingActivityEdge		ResourceInteraction	\$Control[i].ResourceInteraction.realizingActivityEdge
ResourceInteraction.realizingConnector		ResourceInteraction	\$Control[i].ResourceInteraction.realizingConnector
startBoundaryType	ISO8601DateTime	UPDMElement	\$Control[i].startBoundaryType
URI	String	UPDMElement	\$Control[i].URI

158. control

A control is a class whose objects manage interactions between collections of objects.

A control class usually has behavior that is specific for one use case, and a control object usually does not outlive the use case realizations in which it participates.

159. ControlOperator

A control operator is a behavior that is intended to represent an arbitrarily complex logical operator that can be used to enable and disable other actions. When this stereotype is applied to behaviors, the behavior takes control values as inputs or provides them as outputs, that is, it treats control as data. When this stereotype is not applied, the behavior may not have a parameter typed by ControlValue. This stereotype also applies to operations with the same semantics.

160. ControlValue

The ControlValue enumeration is a type for treating control values as data and for UML control pins. It can be used as the type of behavior and operation parameters, object nodes, and attributes, and so on. The possible runtime values are given as enumeration literals. Modelers can extend the enumeration with additional literals, such as suspend, resume, with their own semantics.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
disable	Enumeration Literal	ControlValue	\$ControlValue[i].disable
enable	Enumeration Literal	ControlValue	\$ControlValue[i].enable

161. Copy

A Copy relationship is a dependency between a supplier requirement and a client requirement that specifies that the text of the client requirement is a read-only copy of the text of the supplier requirement.

Base Classifier

- Trace

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getTracedFrom	Requirement	Trace	\$Copy[i].getTracedFrom
sourceContext	Classifier	DirectedRelationshipPropertyPath	\$Copy[i].sourceContext
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	\$Copy[i].sourcePropertyPath

targetContext	Classifier	DirectedRelationshipPropertyPath	<code>\$Copy[i].targetContext</code>
targetPropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$Copy[i].targetPropertyPath</code>

162. Cr

Base Classifier

- [InvisibleStereotype](#)

163. Cr Package

Base Classifier

- [InvisibleStereotype](#)

164. Create

A usage dependency denoting that the client classifier creates instances of the supplier classifier.

Specifies that the designated feature creates an instance of the classifier to which the feature is attached. May be promoted to the Classifier containing the feature.

165. CustomImageHolder

Stereotype used to hold image which is draw in diagram.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Content	String	CustomImageHolder	<code>\$CustomImageHolder[i].Content</code>
Format	String	CustomImageHolder	<code>\$CustomImageHolder[i].Format</code>
Location	String	CustomImageHolder	<code>\$CustomImageHolder[i].Location</code>

166. Customization

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
abbreviation	String	Customization	<code>\$Customization[i].abbreviation</code>

additionalContent	derivedPropertySpecification	Customization	<code>\$Customization[i].additionalContent</code>
allowedDragAndDrops	DragAndDropSpecification	Customization	<code>\$Customization[i].allowedDragAndDrops</code>
allowedRelationships	Class	Customization	<code>\$Customization[i].allowedRelationships</code>
applyToSource	Stereotype	Customization	<code>\$Customization[i].applyToSource</code>
applyToTarget	Stereotype	Customization	<code>\$Customization[i].applyToTarget</code>
category	String	Customization	<code>\$Customization[i].category</code>
checkSpelling	String	Customization	<code>\$Customization[i].checkSpelling</code>
customizationTarget	Class	Customization	<code>\$Customization[i].customizationTarget</code>
defaultShapeSize	int	Customization	<code>\$Customization[i].defaultShapeSize</code>
disallowedRelationships	Class	Customization	<code>\$Customization[i].disallowedRelationships</code>
doNotSuggestAsType	Boolean	Customization	<code>\$Customization[i].doNotSuggestAsType</code>
doNotSuggestNameAutoCompletion	Boolean	Customization	<code>\$Customization[i].doNotSuggestNameAutoCompletion</code>
helpID	String	Customization	<code>\$Customization[i].helpID</code>
hiddenOwnedDiagrams	String	Customization	<code>\$Customization[i].hiddenOwnedDiagrams</code>
hiddenOwnedTypes	Class	Customization	<code>\$Customization[i].hiddenOwnedTypes</code>
hideMetatype	boolean	Customization	<code>\$Customization[i].hideMetatype</code>
inShortcutMenu	Property	Customization	<code>\$Customization[i].inShortcutMenu</code>
keyword	String	Customization	<code>\$Customization[i].keyword</code>
multiLineTextProperties	Property	Customization	<code>\$Customization[i].multiLineTextProperties</code>
possibleOwners	Class	Customization	<code>\$Customization[i].possibleOwners</code>
preferredMetatype	Class	Customization	<code>\$Customization[i].preferredMetatype</code>
quickApplyingFor	Class	Customization	<code>\$Customization[i].quickApplyingFor</code>
representationText	String	Customization	<code>\$Customization[i].representationText</code>
showPropertiesWhenNotApplied	Boolean	Customization	<code>\$Customization[i].showPropertiesWhenNotApplied</code>
showPropertiesWhenNotAppliedLimitedByElementType	Classifier	Customization	<code>\$Customization[i].showPropertiesWhenNotAppliedLimitedByElementType</code>
showPropertiesWhenNotAppliedLimitedByProfileApplication	boolean	Customization	<code>\$Customization[i].showPropertiesWhenNotAppliedLimitedByProfileApplication</code>
standardExpertConfiguration	String	Customization	<code>\$Customization[i].standardExpertConfiguration</code>
subElementContentsIncluded	SubcontentsKind	Customization	<code>\$Customization[i].subElementContentsIncluded</code>
suggestedOwnedDiagrams	String	Customization	<code>\$Customization[i].suggestedOwnedDiagrams</code>
suggestedOwnedTypes	Class	Customization	<code>\$Customization[i].suggestedOwnedTypes</code>
superTypes	Element	Customization	<code>\$Customization[i].superTypes</code>
symbolStandardExpertConfiguration	String	Customization	<code>\$Customization[i].symbolStandardExpertConfiguration</code>
typesForSource	Class	Customization	<code>\$Customization[i].typesForSource</code>
typesForTarget	Class	Customization	<code>\$Customization[i].typesForTarget</code>
usedUMLProperties	String	Customization	<code>\$Customization[i].usedUMLProperties</code>

167. CustomizationGroupNames

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Connection Rules	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].Connection Rules</code>
Content	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].Content</code>
General	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].General</code>
Model Initialization	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].Model Initialization</code>
Naming	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].Naming</code>
OwnedElements	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].OwnedElements</code>
Properties	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].Properties</code>
Symbol	Enumeration Literal	CustomizationGroupNames	<code>\$CustomizationGroupNames[i].Symbol</code>

168. CV-1

Base Classifier

- [InvisibleStereotype](#)

169. CV-1 Package

Base Classifier

- [InvisibleStereotype](#)

170. CV-2

Base Classifier

- [InvisibleStereotype](#)

171. CV-2 Package

Base Classifier

- [InvisibleStereotype](#)

172. CV-3

Base Classifier

- [InvisibleStereotype](#)

173. CV-3 Package

Base Classifier

- [InvisibleStereotype](#)

174. CV-4

Base Classifier

- [InvisibleStereotype](#)

175. CV-4 Package

Base Classifier

- [InvisibleStereotype](#)

176. CV-5

CV-5 Capability to Organizational Development Mapping (CV-5) shows the planned capability deployment and interconnection for a particular Capability Phase.

*Click **Add Rows** button to select Actual Organizational Resources. Selected Actual Organizational Resources shall be added as table rows.*

*Click **Add/Remove Columns** button to select or deselect Capabilities displayed as Columns in the table.*

Table cells are allowed to edit in the table. System Resources can be added to the table as cell contents. Added resources will be marked as exhibiting Capability represented by column and used by Actual Organizational Resource represented by Row in the particular Enterprise time frame.

Rows (Actual Organizational Resources) can be removed from model or only from table, can be ordered, exported to the CSV or HTML and CV-5 Spreadsheet Report can be generated.

More Actions are available by clicking right mouse button on a cell

Base Classifier

- [InvisibleStereotype](#)

177. CV-5 Package

Base Classifier

- [InvisibleStereotype](#)

178. CV-5 Report

Base Classifier

- [InvisibleStereotype](#)

179. CV-6

Capability to Operational Activities Mapping (CV-6) describes the mapping between the capabilities required and the operational activities that those capabilities support.

The Rows of this matrix are Capabilities and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Capabilities);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Operational Activities maps to Capabilities using "Activity Part of Capability" relationship.

To map Operational Activity to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

180. CV-6 Package**Base Classifier**

- [InvisibleStereotype](#)

181. CV-7

Capability to Services Mapping (CV-7) depicts mapping between the capabilities and the services that these capabilities enable.

The Rows of this matrix are Service Accesses and the Columns are Capabilities.

To build the Matrix:

- 1. Specify Rows scope (Service Accesses);*
- 2. Specify Columns scope (Capabilities);*
- 3. Click "Refresh" button.*

Service Accesses expose Capabilities using "Capability of Performer" relationship.

To map Service Access to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

182. CV-7 Package**Base Classifier**

- [InvisibleStereotype](#)

183. D1

The Master Data (D1) provides definitions of all terms used throughout the architectural data. To fill in D1 table, UPDM elements have to be added to it.

*Click **Add Element** button to create a new or to add an existing element to the table.*

"Name", "Definition", "Alias", "Same As", "Documentation", and wide range of column cells are allowed to edit in the table. "UPDM Type", "UML Metatype", "SysML Type", "BPMN Type" cells are read only.

Rows (UPDM Elements) can be removed from the model or only from the table, can be ordered, and exported to the CSV or HTML. Four kind of reports can be printed reflecting the data shown in the table.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	D1	\$D1[i].hideColumns

184. D1 Package

Base Classifier

- [InvisibleStereotype](#)

185. D1 Report

Base Classifier

- [InvisibleStereotype](#)

186. D1 Tabular Report

Base Classifier

- [InvisibleStereotype](#)

187. D2

Base Classifier

- [InvisibleStereotype](#)

188. D2 Package

Base Classifier

- [InvisibleStereotype](#)

189. DARS Template

Base Classifier

- [ArchitecturalDescription](#)
- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualCost	float	DARS Template	<code>\$DARSTemplate[i].actualCost</code>
actualLevelOfEffort	String	DARS Template	<code>\$DARSTemplate[i].actualLevelOfEffort</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$DARSTemplate[i].actualPropertySet</code>
analysts	ActualOrganization	DARS Template	<code>\$DARSTemplate[i].analysts</code>
approvalAuthority	String	ArchitecturalDescription	<code>\$DARSTemplate[i].approvalAuthority</code>
approvalDate	ISO8601DateTime	DARS Template	<code>\$DARSTemplate[i].approvalDate</code>
approvalStatus	ApprovalStatus	DARS Template	<code>\$DARSTemplate[i].approvalStatus</code>
architect	String	ArchitecturalDescription	<code>\$DARSTemplate[i].architect</code>
ArchitecturalDescription.architectureFramework		ArchitecturalDescription	<code>\$DARSTemplate[i].ArchitecturalDescription.architectureFramework</code>
architectureFramework	ArchitectureFrameworkKind	ArchitecturalDescription	<code>\$DARSTemplate[i].architectureFramework</code>
architectureName	String	DARS Template	<code>\$DARSTemplate[i].architectureName</code>

architectureURL	String	DARS Template	\$DARSTemplate[i].architectureURL
assumptionAndConstraint	String	ArchitecturalDescription	\$DARSTemplate[i].assumptionAndConstraint
authoritativeSourcesForGuidance	String	DARS Template	\$DARSTemplate[i].authoritativeSourcesForGuidance
authoritativeSourcesForITStandard	String	DARS Template	\$DARSTemplate[i].authoritativeSourcesForITStandard
communitiesOfInterest	String	DARS Template	\$DARSTemplate[i].communitiesOfInterest
completionStatus	CompletionStatus	DARS Template	\$DARSTemplate[i].completionStatus
conceptOfOperations	String	DARS Template	\$DARSTemplate[i].conceptOfOperations
conclusions	Findings	DARS Template	\$DARSTemplate[i].conclusions
conformsTo	Standard	UPDMElement	\$DARSTemplate[i].conformsTo
constraints	Findings	DARS Template	\$DARSTemplate[i].constraints
contributingOrganizations	ActualOrganization	DARS Template	\$DARSTemplate[i].contributingOrganizations
creatingOrganization	String	ArchitecturalDescription	\$DARSTemplate[i].creatingOrganization
creatorEmail	String	DARS Template	\$DARSTemplate[i].creatorEmail
creatorFirstName	String	DARS Template	\$DARSTemplate[i].creatorFirstName
creatorLastName	String	DARS Template	\$DARSTemplate[i].creatorLastName
creatorMI	String	DARS Template	\$DARSTemplate[i].creatorMI
creatorPhone	String	DARS Template	\$DARSTemplate[i].creatorPhone
dataRestrictions	DataRestrictions	DARS Template	\$DARSTemplate[i].dataRestrictions
dateCompleted	String	ArchitecturalDescription	\$DARSTemplate[i].dateCompleted
decisionMakers	ActualOrganization	DARS Template	\$DARSTemplate[i].decisionMakers
decisions	String	DARS Template	\$DARSTemplate[i].decisions
description	String	DARS Template	\$DARSTemplate[i].description
documentAccessLevel	DocumentAccessLevel	DARS Template	\$DARSTemplate[i].documentAccessLevel
endBoundaryType	ISO8601DateTime	UPDMElement	\$DARSTemplate[i].endBoundaryType
environmentalConditionDescriptions	String	DARS Template	\$DARSTemplate[i].environmentalConditionDescriptions
environmentalConditionNames	LocationType	DARS Template	\$DARSTemplate[i].environmentalConditionNames
estimateCost	float	DARS Template	\$DARSTemplate[i].estimateCost
estimateLevelOfEffort	String	DARS Template	\$DARSTemplate[i].estimateLevelOfEffort
fileFormatUsed	String	DARS Template	\$DARSTemplate[i].fileFormatUsed
geographicalAreaName	ActualLocation	DARS Template	\$DARSTemplate[i].geographicalAreaName
granularityLevel	GranularityLevel	DARS Template	\$DARSTemplate[i].granularityLevel
guidanceReferences	String	DARS Template	\$DARSTemplate[i].guidanceReferences
implementability	Implementability	DARS Template	\$DARSTemplate[i].implementability
infoAssuranceRequirement	String	DARS Template	\$DARSTemplate[i].infoAssuranceRequirement
infoAssuranceThreatRequirement	String	DARS Template	\$DARSTemplate[i].infoAssuranceThreatRequirement
issues	Findings	DARS Template	\$DARSTemplate[i].issues

jointCapabilityArea	String	DARS Template	\$DARSTemplate[i].jointCapabilityArea
jointPotentialDesignator	JointPotentialDesignator	DARS Template	\$DARSTemplate[i].jointPotentialDesignator
lastModifiedDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].lastModifiedDate
methodologiesUsed	String	DARS Template	\$DARSTemplate[i].methodologiesUsed
missionName	Mission	DARS Template	\$DARSTemplate[i].missionName
objective	String	DARS Template	\$DARSTemplate[i].objective
organizationsInvolved	ActualOrganization	DARS Template	\$DARSTemplate[i].organizationsInvolved
primaryUse	String	DARS Template	\$DARSTemplate[i].primaryUse
projectEndDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].projectEndDate
projectStartDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].projectStartDate
propertySet	PropertySet	UPDMElement	\$DARSTemplate[i].propertySet
proposedActions	String	DARS Template	\$DARSTemplate[i].proposedActions
proprietaryInformation	ProprietaryInformation	DARS Template	\$DARSTemplate[i].proprietaryInformation
purpose	String	ArchitecturalDescription	\$DARSTemplate[i].purpose
recommendations	String	ArchitecturalDescription	\$DARSTemplate[i].recommendations
recommendations	Findings	DARS Template	\$DARSTemplate[i].recommendations
registrationDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].registrationDate
relatedArchitectures	String	DARS Template	\$DARSTemplate[i].relatedArchitectures
releasability	Releasability	DARS Template	\$DARSTemplate[i].releasability
scenarios	String	DARS Template	\$DARSTemplate[i].scenarios
securityClassification	SecurityClassification	DARS Template	\$DARSTemplate[i].securityClassification
specialHandlingInstructions	SpecialHandlingInstructions	DARS Template	\$DARSTemplate[i].specialHandlingInstructions
startBoundaryType	ISO8601DateTime	UPDMElement	\$DARSTemplate[i].startBoundaryType
summaryOfFindings	String	ArchitecturalDescription	\$DARSTemplate[i].summaryOfFindings
suspendedDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].suspendedDate
taskingAgent	ActualOrganization	DARS Template	\$DARSTemplate[i].taskingAgent
temporalScope	TemporalScope	DARS Template	\$DARSTemplate[i].temporalScope
threatDescription	String	DARS Template	\$DARSTemplate[i].threatDescription
threatNames	String	DARS Template	\$DARSTemplate[i].threatNames
timeFrameEndDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].timeFrameEndDate
timeFrameName	String	DARS Template	\$DARSTemplate[i].timeFrameName
timeFrameStartDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].timeFrameStartDate
toBe	Boolean	ArchitecturalDescription	\$DARSTemplate[i].toBe
toolsUsed	String	ArchitecturalDescription	\$DARSTemplate[i].toolsUsed
URI	String	UPDMElement	\$DARSTemplate[i].URI
validatingOrganization	ActualOrganization	DARS Template	\$DARSTemplate[i].validatingOrganization
validationDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].validationDate
validUntilDate	ISO8601DateTime	DARS Template	\$DARSTemplate[i].validUntilDate
version	Integer	DARS Template	\$DARSTemplate[i].version
viewpoint	String	ArchitecturalDescription	\$DARSTemplate[i].viewpoint

viewpoint	String	DARS Template	<code>\$DARSTemplate[i].viewpoint</code>
views	View	ArchitecturalDescription	<code>\$DARSTemplate[i].views</code>
vulnerabilities	Findings	DARS Template	<code>\$DARSTemplate[i].vulnerabilities</code>

190. Data and Information Viewpoint

Base Classifier

- [InvisibleStereotype](#)

191. DataModel

MODAF: A structural specification of data, showing classifications of data elements and relationships between them.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$DataModel[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$DataModel[i].conformsTo</code>
DataModel.ownedElement		DataModel	<code>\$DataModel[i].DataModel.ownedElement</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$DataModel[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$DataModel[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$DataModel[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$DataModel[i].URI</code>

192. DataRestrictions

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
formerly restricted data	Enumeration Literal	DataRestrictions	<code>\$DataRestrictions[i].formerly restricted data</code>
not known	Enumeration Literal	DataRestrictions	<code>\$DataRestrictions[i].not known</code>
not restricted data	Enumeration Literal	DataRestrictions	<code>\$DataRestrictions[i].not restricted data</code>
not specified	Enumeration Literal	DataRestrictions	<code>\$DataRestrictions[i].not specified</code>
restricted data	Enumeration Literal	DataRestrictions	<code>\$DataRestrictions[i].restricted data</code>

193. debugIcon

Base Classifier

- [imaged](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
highlightColor	String	imaged	\$debugIcon[i].highlightColor

194. Definition

MODAF: A definition of an element in the architecture.

DoDAF:NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Definition[i].actualPropertySet
author	String	Definition	\$Definition[i].author
conformsTo	Standard	UPDMElement	\$Definition[i].conformsTo
Definition.annotatedElement		Definition	\$Definition[i].Definition.annotatedElement
endBoundaryType	ISO8601DateTime	UPDMElement	\$Definition[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$Definition[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$Definition[i].startBoundaryType
URI	String	UPDMElement	\$Definition[i].URI

195. definition

196. delegate

197. DependencyMatrix

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
columnCollapsedNodes	String	DependencyMatrix	<code>\$DependencyMatrix[i].columnCollapsedNodes</code>
columnCustomOrder	String	DependencyMatrix	<code>\$DependencyMatrix[i].columnCustomOrder</code>
columnHeaderHeight	int	DependencyMatrix	<code>\$DependencyMatrix[i].columnHeaderHeight</code>
columnOwnerDisplayMode	OwnerDisplayMode	DependencyMatrix	<code>\$DependencyMatrix[i].columnOwnerDisplayMode</code>
columnSortingMode	SortingMode	DependencyMatrix	<code>\$DependencyMatrix[i].columnSortingMode</code>
columnTextDirection	TextDirection	DependencyMatrix	<code>\$DependencyMatrix[i].columnTextDirection</code>
dependencyCriteria	StructuredExpression	DependencyMatrix	<code>\$DependencyMatrix[i].dependencyCriteria</code>
descriptionArea	String	DependencyMatrix	<code>\$DependencyMatrix[i].descriptionArea</code>
direction	Direction	DependencyMatrix	<code>\$DependencyMatrix[i].direction</code>
hideDependencyCriteria	boolean	DependencyMatrix	<code>\$DependencyMatrix[i].hideDependencyCriteria</code>
hideScope	boolean	DependencyMatrix	<code>\$DependencyMatrix[i].hideScope</code>
hideTypes	boolean	DependencyMatrix	<code>\$DependencyMatrix[i].hideTypes</code>
legendLocation	LegendLocation	DependencyMatrix	<code>\$DependencyMatrix[i].legendLocation</code>
readOnly	boolean	DependencyMatrix	<code>\$DependencyMatrix[i].readOnly</code>
rowCollapsedNodes	String	DependencyMatrix	<code>\$DependencyMatrix[i].rowCollapsedNodes</code>
rowCustomOrder	String	DependencyMatrix	<code>\$DependencyMatrix[i].rowCustomOrder</code>
rowHeaderWidth	int	DependencyMatrix	<code>\$DependencyMatrix[i].rowHeaderWidth</code>
rowOwnerDisplayMode	OwnerDisplayMode	DependencyMatrix	<code>\$DependencyMatrix[i].rowOwnerDisplayMode</code>
rowSortingMode	SortingMode	DependencyMatrix	<code>\$DependencyMatrix[i].rowSortingMode</code>
showElements	RelationOption	DependencyMatrix	<code>\$DependencyMatrix[i].showElements</code>
showInnerDependencies	boolean	DependencyMatrix	<code>\$DependencyMatrix[i].showInnerDependencies</code>
suppressCriteriaArea	boolean	DependencyMatrix	<code>\$DependencyMatrix[i].suppressCriteriaArea</code>
takeWholeModelAsScope	boolean	DependencyMatrix	<code>\$DependencyMatrix[i].takeWholeModelAsScope</code>

198. Deployed Resources Viewpoint

Base Classifier

- [InvisibleStereotype](#)

199. DeployedMilestone

MODAF: Asserts that an ActualOrganisationResource started to use, or is slated to start using a CapabilityConfiguration from a specific point in time. --This is used to describe capabilities going into service with specific organisations or posts.

DoDAF: NA

Base Classifier

- [ActualProjectMilestone](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualProjectMilestone.classifier		ActualProjectMilestone	<code>\$DeployedMilestone[i].ActualProjectMilestone.classifier</code>
ActualProjectMilestone.slot		ActualProjectMilestone	<code>\$DeployedMilestone[i].ActualProjectMilestone.slot</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$DeployedMilestone[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$DeployedMilestone[i].conformsTo</code>
date	ISO8601DateTime	ActualProjectMilestone	<code>\$DeployedMilestone[i].date</code>
description	String	ActualProjectMilestone	<code>\$DeployedMilestone[i].description</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$DeployedMilestone[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$DeployedMilestone[i].propertySet</code>
resource	SystemResource	ActualProjectMilestone	<code>\$DeployedMilestone[i].resource</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$DeployedMilestone[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$DeployedMilestone[i].URI</code>
usedBy	ActualOrganizationalResource	DeployedMilestone	<code>\$DeployedMilestone[i].usedBy</code>

200. deploymentView

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
deploymentViewID	int	deploymentView	<code>\$deploymentView[i].deploymentViewID</code>

201. deprecated

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
deprecatedReason	String	deprecated	<code>\$deprecated[i].deprecatedReason</code>

202. Derive

Specifies a derivation relationship among model elements that are usually, but not necessarily, of the same type. A derived dependency specifies that the client may be

computed from the supplier. The mapping specifies the computation. The client may be implemented for design reasons, such as efficiency, even though it is logically redundant.

203. derivedPropertiesSuite

204. derivedPropertySpecification

Stereotype to define derived property. Property owned by DSL customization class should be stereotyped with `derivedPropertySpecification` in order to be included in MagicDraw as derived property. `derivedPropertySpecification` element has one or many expression to calculate derived property in different languages. Derived properties result is union of results of defined expressions.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
expression	StructuredExpression	derivedPropertySpecification	<code>\$derivedPropertySpecification[i].expression</code>
isReadOnly	Boolean	derivedPropertySpecification	<code>\$derivedPropertySpecification[i].isReadOnly</code>
valueSetter	String	derivedPropertySpecification	<code>\$derivedPropertySpecification[i].valueSetter</code>

205. DeriveReq

A `DeriveReq` relationship is a dependency between two requirements in which a client requirement can be derived from the supplier requirement. As with other dependencies, the arrow direction points from the derived (client) requirement to the (supplier) requirement from which it is derived.

Base Classifier

- [Trace](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getTracedFrom	Requirement	Trace	<code>\$DeriveReq[i].getTracedFrom</code>
sourceContext	Classifier	DirectedRelationshipPropertyPath	<code>\$DeriveReq[i].sourceContext</code>
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$DeriveReq[i].sourcePropertyPath</code>
targetContext	Classifier	DirectedRelationshipPropertyPath	<code>\$DeriveReq[i].targetContext</code>
targetPropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$DeriveReq[i].targetPropertyPath</code>

206. designConstraint

Requirement that specifies a constraint on the implementation of the system or system part, such as the system must use a commercial off the shelf component.

Base Classifier

- [extendedRequirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	\$designConstraint[i].Derived
DerivedFrom	Requirement	Requirement	\$designConstraint[i].DerivedFrom
Id	String	Requirement	\$designConstraint[i].Id
Master	Requirement	Requirement	\$designConstraint[i].Master
RefinedBy	NamedElement	Requirement	\$designConstraint[i].RefinedBy
risk	RiskKind	extendedRequirement	\$designConstraint[i].risk
SatisfiedBy	NamedElement	Requirement	\$designConstraint[i].SatisfiedBy
source	String	extendedRequirement	\$designConstraint[i].source
Text	String	Requirement	\$designConstraint[i].Text
TracedTo	NamedElement	Requirement	\$designConstraint[i].TracedTo
VerifiedBy	NamedElement	Requirement	\$designConstraint[i].VerifiedBy
verifyMethod	VerificationMethodKind	extendedRequirement	\$designConstraint[i].verifyMethod

207. designModel

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
designModelID	int	designModel	\$designModel[i].designModelID

208. DesignRule

A design rule is a solution to a problem in a specific context with the following characteristics:

- belongs to a problem domain,
- packages knowledge in a reusable form,
- standardize solutions to design problems within NBD,
- gives value to the re-user.

Base Classifier

- [Rule](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$DesignRule[i].actualPropertySet

analysis	String	DesignRule	\$DesignRule[i].analysis
conformsTo	Standard	UPDMElement	\$DesignRule[i].conformsTo
consequence	String	DesignRule	\$DesignRule[i].consequence
context	String	DesignRule	\$DesignRule[i].context
date	ISO8601DateTime	DesignRule	\$DesignRule[i].date
DesignRule.ruleKind		DesignRule	\$DesignRule[i].DesignRule.ruleKind
endBoundaryType	ISO8601DateTime	UPDMElement	\$DesignRule[i].endBoundaryType
identifier	String	DesignRule	\$DesignRule[i].identifier
metaData	String	DesignRule	\$DesignRule[i].metaData
principles	String	DesignRule	\$DesignRule[i].principles
problem	String	DesignRule	\$DesignRule[i].problem
propertySet	PropertySet	UPDMElement	\$DesignRule[i].propertySet
ruleKind	RuleKind	Rule	\$DesignRule[i].ruleKind
solution	Element	DesignRule	\$DesignRule[i].solution
startBoundaryType	ISO8601DateTime	UPDMElement	\$DesignRule[i].startBoundaryType
status	DevelopmentStatus	DesignRule	\$DesignRule[i].status
URI	String	UPDMElement	\$DesignRule[i].URI
version	String	DesignRule	\$DesignRule[i].version

209. DesiredEffect

MODAF:NA

DoDAF:A desired state of a Resource.

Base Classifier

- UPDMElement

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$DesiredEffect[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$DesiredEffect[i].conformsTo
DesiredEffect.client		DesiredEffect	\$DesiredEffect[i].DesiredEffect.client
DesiredEffect.supplier		DesiredEffect	\$DesiredEffect[i].DesiredEffect.supplier
desiredResourceFutureState	State	DesiredEffect	\$DesiredEffect[i].desiredResourceFutureState
endBoundaryType	ISO8601DateTime	UPDMElement	\$DesiredEffect[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$DesiredEffect[i].propertySet
providedMOE	ActualPropertySet	DesiredEffect	\$DesiredEffect[i].providedMOE
startBoundaryType	ISO8601DateTime	UPDMElement	\$DesiredEffect[i].startBoundaryType
URI	String	UPDMElement	\$DesiredEffect[i].URI

210. DesiredState

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$DesiredState[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$DesiredState[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$DesiredState[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$DesiredState[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$DesiredState[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$DesiredState[i].URI</code>

211. Desirer

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Desirer[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Desirer[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Desirer[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Desirer[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Desirer[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Desirer[i].URI</code>

212. Destroy

Specifies that the designated feature destroys an instance of the classifier to which the feature is attached. May be promoted to the classifier containing the feature.

213. destructor

214. Details

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Details[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Details[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Details[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Details[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Details[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Details[i].URI</code>

215. DevelopmentStatus

Enumeration of development statuses, used to support the status tag of the DesignRule stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Draft	Enumeration Literal	DevelopmentStatus	<code>\$DevelopmentStatus[i].Draft</code>
Identified	Enumeration Literal	DevelopmentStatus	<code>\$DevelopmentStatus[i].Identified</code>
Obsolete	Enumeration Literal	DevelopmentStatus	<code>\$DevelopmentStatus[i].Obsolete</code>
Proposal	Enumeration Literal	DevelopmentStatus	<code>\$DevelopmentStatus[i].Proposal</code>
Rejected	Enumeration Literal	DevelopmentStatus	<code>\$DevelopmentStatus[i].Rejected</code>
Verified	Enumeration Literal	DevelopmentStatus	<code>\$DevelopmentStatus[i].Verified</code>

216. Diagram Description

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Completion status	String	Diagram Description	<code>\$DiagramDescription[i].Completion status</code>
Description	String	Diagram Description	<code>\$DiagramDescription[i].Description</code>
Reference	Element	Diagram Description	<code>\$DiagramDescription[i].Reference</code>
Version	String	Diagram Description	<code>\$DiagramDescription[i].Version</code>

217. DiagramCollectingMethod

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

Context	Enumeration Literal	DiagramCollectingMethod	<code>\$DiagramCollectingMethod[i].Context</code>
Owner	Enumeration Literal	DiagramCollectingMethod	<code>\$DiagramCollectingMethod[i].Owner</code>
Self	Enumeration Literal	DiagramCollectingMethod	<code>\$DiagramCollectingMethod[i].Self</code>

218. DiagramInfo

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Author	String	DiagramInfo	<code>\$DiagramInfo[i].Author</code>
Creation date	date	DiagramInfo	<code>\$DiagramInfo[i].Creation date</code>
Last modified by	String	DiagramInfo	<code>\$DiagramInfo[i].Last modified by</code>
Modification date	date	DiagramInfo	<code>\$DiagramInfo[i].Modification date</code>

219. DiagramLegend

Base Classifier

- [InvisibleStereotype](#)

220. Diagrams

Base Classifier

- [Panel](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
diagramCollectingMethod	DiagramCollectingMethod	Diagrams	<code>\$Diagrams[i].diagramCollectingMethod</code>
diagramTypes	String	Diagrams	<code>\$Diagrams[i].diagramTypes</code>

221. DiagramsDefaultName

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
diagramName	String	DiagramsDefaultName	<code>\$DiagramsDefaultName[i].diagramName</code>

diagramType	String	DiagramsDefaultName	<code>\$DiagramsDefaultName[i].diagramType</code>
-------------	--------	-------------------------------------	---

222. DiagramTable

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
additionalElements	Element	DiagramTable	<code>\$DiagramTable[i].additionalElements</code>
autoResizeOn	Boolean	DiagramTable	<code>\$DiagramTable[i].autoResizeOn</code>
columnElements	String	DiagramTable	<code>\$DiagramTable[i].columnElements</code>
columnIds	String	DiagramTable	<code>\$DiagramTable[i].columnIds</code>
columnWidth	Integer	DiagramTable	<code>\$DiagramTable[i].columnWidth</code>
customColumns	String	DiagramTable	<code>\$DiagramTable[i].customColumns</code>
defaultRowElementsOwner	Element	DiagramTable	<code>\$DiagramTable[i].defaultRowElementsOwner</code>
excludedElements	Element	DiagramTable	<code>\$DiagramTable[i].excludedElements</code>
hideColumns	String	DiagramTable	<code>\$DiagramTable[i].hideColumns</code>
query	StructuredExpression	DiagramTable	<code>\$DiagramTable[i].query</code>
rowElements	String	DiagramTable	<code>\$DiagramTable[i].rowElements</code>
rowElementType	Element	DiagramTable	<code>\$DiagramTable[i].rowElementType</code>
rowFilters	String	DiagramTable	<code>\$DiagramTable[i].rowFilters</code>
rowHeight	Integer	DiagramTable	<code>\$DiagramTable[i].rowHeight</code>
scope	Element	DiagramTable	<code>\$DiagramTable[i].scope</code>
showDetailedColumnName	Boolean	DiagramTable	<code>\$DiagramTable[i].showDetailedColumnName</code>
showFullPath	Boolean	DiagramTable	<code>\$DiagramTable[i].showFullPath</code>
sort	String	DiagramTable	<code>\$DiagramTable[i].sort</code>
typesIncludeSubtypes	boolean	DiagramTable	<code>\$DiagramTable[i].typesIncludeSubtypes</code>

223. diagramUsage

SysML also introduces the concept of a diagram usage. This represents a unique usage of a particular diagram type, such as a context diagram as a usage of an block definition diagram, internal block diagram, or use case diagram. The diagram usage can be identified in the header above the diagramKind as «diagramUsage».

224. DirectedFeature

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
featureDirection	FeatureDirection	DirectedFeature	<code>\$DirectedFeature[i].featureDirection</code>

225. DirectedRelationshipPropertyPath

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
sourceContext	Classifier	DirectedRelationshipPropertyPath	<code>\$DirectedRelationshipPropertyPath[i].sourceContext</code>
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$DirectedRelationshipPropertyPath[i].sourcePropertyPath</code>
targetContext	Classifier	DirectedRelationshipPropertyPath	<code>\$DirectedRelationshipPropertyPath[i].targetContext</code>
targetPropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$DirectedRelationshipPropertyPath[i].targetPropertyPath</code>

226. Direction

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Both	Enumeration Literal	Direction	<code>\$Direction[i].Both</code>
Column to row	Enumeration Literal	Direction	<code>\$Direction[i].Column to row</code>
Row to column	Enumeration Literal	Direction	<code>\$Direction[i].Row to column</code>

227. Discrete

Discrete rate is a special case of rate of flow (see Rate) where the increment of time between items is non-zero.

Base Classifier

- [Rate](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
rate	InstanceSpecification	Rate	<code>\$Discrete[i].rate</code>

228. DistributedProperty

DistributedProperty is a stereotype of *Property* used to apply a probability distribution to the values of the property. Specific distributions should be defined as subclasses of the *DistributedProperty* stereotype with the operands of the distributions represented by properties of those stereotype subclasses.

229. DIV-1

Base Classifier

- [InvisibleStereotype](#)

230. DIV-1 Package

Base Classifier

- [InvisibleStereotype](#)

231. DIV-2

Base Classifier

- [InvisibleStereotype](#)

232. DIV-2 Package

Base Classifier

- [InvisibleStereotype](#)

233. DIV-3

Base Classifier

- [InvisibleStereotype](#)

234. DIV-3 Package

Base Classifier

- [InvisibleStereotype](#)

235. DLODSegment

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Critical Issues	Enumeration Literal	DLODSegment	\$DLODSegment[i].Critical Issues
DLOD Absent	Enumeration Literal	DLODSegment	\$DLODSegment[i].DLOD Absent
Manageable Issues	Enumeration Literal	DLODSegment	\$DLODSegment[i].Manageable Issues
No Outstanding Issues	Enumeration Literal	DLODSegment	\$DLODSegment[i].No Outstanding Issues
Not Required	Enumeration Literal	DLODSegment	\$DLODSegment[i].Not Required

236. DLODStatus**Base Classifier**

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
doctrine/concepts	DLODSegment	DLODStatus	\$DLODStatus[i].doctrine/concepts
equipment	DLODSegment	DLODStatus	\$DLODStatus[i].equipment
information	DLODSegment	DLODStatus	\$DLODStatus[i].information
infrastructure	DLODSegment	DLODStatus	\$DLODStatus[i].infrastructure
logistics	DLODSegment	DLODStatus	\$DLODStatus[i].logistics
organization	DLODSegment	DLODStatus	\$DLODStatus[i].organization
personnel	DLODSegment	DLODStatus	\$DLODStatus[i].personnel
training	DLODSegment	DLODStatus	\$DLODStatus[i].training

237. Document

A generic file that is not a «source» file or «executable».

Subclass of «file».

Base Classifier

- [File](#)

238. DocumentAccessLevel

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
DARS public	Enumeration Literal	DocumentAccessLevel	<code>\$DocumentAccessLevel[i].DARS public</code>
general public	Enumeration Literal	DocumentAccessLevel	<code>\$DocumentAccessLevel[i].general public</code>
private	Enumeration Literal	DocumentAccessLevel	<code>\$DocumentAccessLevel[i].private</code>
protected	Enumeration Literal	DocumentAccessLevel	<code>\$DocumentAccessLevel[i].protected</code>

239. DoDAF 2.0 All Views Report

Base Classifier

- [InvisibleStereotype](#)

240. DoDAF 2.0 AV-1 Report

Base Classifier

- [InvisibleStereotype](#)

241. DoDAF 2.0 AV-2 Report

Base Classifier

- [InvisibleStereotype](#)

242. DoDAF 2.0 AV-2 Tabular Report

Base Classifier

- [InvisibleStereotype](#)

243. DoDAF 2.0 OV-3 Report

Base Classifier

- [InvisibleStereotype](#)

244. DoDAF 2.0 OV-3 Role Based Report**Base Classifier**

- [InvisibleStereotype](#)

245. DoDAF 2.0 OV-6a Report**Base Classifier**

- [InvisibleStereotype](#)

246. DoDAF 2.0 SV-10a Report**Base Classifier**

- [InvisibleStereotype](#)

247. DoDAF 2.0 SV-6 Report**Base Classifier**

- [InvisibleStereotype](#)

248. DoDAF 2.0 SV-6 Role Based Report**Base Classifier**

- [InvisibleStereotype](#)

249. DoDAF 2.0 SV-7 Report**Base Classifier**

- [InvisibleStereotype](#)

250. DoDAF 2.0 SV-8 Report

Base Classifier

- [InvisibleStereotype](#)

251. DoDAFProperties

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
appliesFor	Element	DoDAFProperties	\$DoDAFProperties[i].appliesFor

252. Domain

A *Domain block* represents an entity, a concept, a location, or a person from the real-world domain. A domain block is part of the system knowledge.

Base Classifier

- [Block](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	\$Domain[i].isEncapsulated

253. Dr

Dr Deployment Schedule shows deployment of capability configurations to specific organizations during a specific Enterprise Phase.

Click **Add Rows** button to select Actual Organizational Resources. Selected Actual Organizational Resources shall be added as table rows.

Click **Add/Remove Columns** button to select or deselect Capabilities displayed as Columns in the table.

Table cells are allowed to edit in the table. System Resources can be added to the table as cells contents. Added resources will be marked as exhibiting Capability represented by column and used by Actual Organizational Resource represented by Row in the particular Enterprise time frame.

Rows (Actual Organization Resources) can be removed from model or only from table, can be ordered, exported to the CSV or HTML and Dr Spreadsheet Report can be generated.

More Actions are available by clicking right mouse button on a cell

Base Classifier

- [InvisibleStereotype](#)
- [StV-5](#)

254. Dr Package

Base Classifier

- [InvisibleStereotype](#)

255. Dr Report

Base Classifier

- [InvisibleStereotype](#)

256. DragAndDropSpecification

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
appendMode	Boolean	DragAndDropSpecification	<code>\$DragAndDropSpecification[i].appendMode</code>
propertyActionResult	String	DragAndDropSpecification	<code>\$DragAndDropSpecification[i].propertyActionResult</code>
relationActionResult	Class	DragAndDropSpecification	<code>\$DragAndDropSpecification[i].relationActionResult</code>
representationText	String	DragAndDropSpecification	<code>\$DragAndDropSpecification[i].representationText</code>
sourceElement	Class	DragAndDropSpecification	<code>\$DragAndDropSpecification[i].sourceElement</code>

257. ED

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ed	String	ED	\$ED[i].ed

258. effbd

Enhanced Functional Flow Block Diagrams (EFFBD) are a widely-used systems engineering diagram, also called a behavior diagram. Most of its functionality is a constrained use of UML activities. EFFBD specifies that the activity conforms to the constraints necessary for EFFBD.

259. ElementGroup

The ElementGroup stereotype provides a lightweight mechanism for grouping various and possibly heterogeneous model elements by extending the capability of comments to refer to multiple annotated elements. For example, it can group elements that are associated with a particular release of the model, have a certain risk level, or are associated with a legacy design.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
allGroups	ElementGroup	ElementGroup	\$ElementGroup[i].allGroups
criterion	String	ElementGroup	\$ElementGroup[i].criterion
criterion	String	ElementGroup	\$ElementGroup[i].criterion
member	Element	ElementGroup	\$ElementGroup[i].member
member	Element	ElementGroup	\$ElementGroup[i].member
name	String	ElementGroup	\$ElementGroup[i].name
orderedMember	Element	ElementGroup	\$ElementGroup[i].orderedMember
size	Integer	ElementGroup	\$ElementGroup[i].size
size	Integer	ElementGroup	\$ElementGroup[i].size

260. ElementPropertyPath

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

propertyPath	Property	ElementPropertyPath	<code>\$ElementPropertyPath[i].propertyPath</code>
--------------	--------------------------	-------------------------------------	--

261. elementsLibrary

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
contents	Element	elementsLibrary	<code>\$elementsLibrary[i].contents</code>
indexElementIDs	String	elementsLibrary	<code>\$elementsLibrary[i].indexElementIDs</code>
indexMetaTypes	String	elementsLibrary	<code>\$elementsLibrary[i].indexMetaTypes</code>
indexNamespaces	String	elementsLibrary	<code>\$elementsLibrary[i].indexNamespaces</code>
indexOwnersIDs	String	elementsLibrary	<code>\$elementsLibrary[i].indexOwnersIDs</code>
indexOwnersModulesIDs	String	elementsLibrary	<code>\$elementsLibrary[i].indexOwnersModulesIDs</code>
indexShortNames	String	elementsLibrary	<code>\$elementsLibrary[i].indexShortNames</code>
structure	treeStructureEnumeration	elementsLibrary	<code>\$elementsLibrary[i].structure</code>

262. elementsLibraryBranch

Base Classifier

- [InvisibleStereotype](#)

263. EndPathMultiplicity

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
lower	Integer	EndPathMultiplicity	<code>\$EndPathMultiplicity[i].lower</code>
upper	UnlimitedNatural	EndPathMultiplicity	<code>\$EndPathMultiplicity[i].upper</code>

264. EnduringTask

MODAF: A type of behaviour recognised by an enterprise as being essential to achieving its goals - i.e. a strategic specification of what the enterprise does.

DoDAF: NA

Base Classifier

- [Process](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EnduringTask[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$EnduringTask[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnduringTask[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$EnduringTask[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnduringTask[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$EnduringTask[i].URI</code>

265. Energy

UPDM: Energy to be exchanged between Nodes.

MODAF: A unit of energy that flows along an EnergyFlow or OperationalActivityEnergyFlow

DoDAF: NA

Base Classifier

- [OperationalExchangeItem](#)
- [ResourceInteractionItem](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$Energy[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Energy[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$Energy[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$Energy[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Energy[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Energy[i].endBoundaryType</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$Energy[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$Energy[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$Energy[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Energy[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Energy[i].URI</code>

266. EnterpriseGoal

MODAF: A specific, required objective of the enterprise that the architecture represents.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EnterpriseGoal[i].actualPropertySet</code>
benefits	String	EnterpriseGoal	<code>\$EnterpriseGoal[i].benefits</code>
conformsTo	Standard	UPDMElement	<code>\$EnterpriseGoal[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnterpriseGoal[i].endBoundaryType</code>
enterprisePhase	EnterprisePhase	EnterpriseGoal	<code>\$EnterpriseGoal[i].enterprisePhase</code>
propertySet	PropertySet	UPDMElement	<code>\$EnterpriseGoal[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnterpriseGoal[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$EnterpriseGoal[i].URI</code>

267. EnterprisePhase

MODAF: A specific, required objective of the enterprise that the architecture represents.

DoDAF: NA

Base Classifier

- [CapableElement](#)
- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EnterprisePhase[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$EnterprisePhase[i].conformsTo</code>
describedBy	ArchitecturalDescription	EnterprisePhase	<code>\$EnterprisePhase[i].describedBy</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnterprisePhase[i].endBoundaryType</code>
endDate	ISO8601DateTime	EnterprisePhase	<code>\$EnterprisePhase[i].endDate</code>
Enterprise from/to		EnterprisePhase	<code>\$EnterprisePhase[i].Enterprise from/to</code>
EnterprisePhase.useCase		EnterprisePhase	<code>\$EnterprisePhase[i].EnterprisePhase.useCase</code>
fulfills	Mission	EnterprisePhase	<code>\$EnterprisePhase[i].fulfills</code>
goals	EnterpriseGoal	EnterprisePhase	<code>\$EnterprisePhase[i].goals</code>
propertySet	PropertySet	UPDMElement	<code>\$EnterprisePhase[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnterprisePhase[i].startBoundaryType</code>
startDate	ISO8601DateTime	EnterprisePhase	<code>\$EnterprisePhase[i].startDate</code>
statementTasks	EnduringTask	EnterprisePhase	<code>\$EnterprisePhase[i].statementTasks</code>
URI	String	UPDMElement	<code>\$EnterprisePhase[i].URI</code>
visions	EnterpriseVision	EnterprisePhase	<code>\$EnterprisePhase[i].visions</code>

268. EnterpriseVision

MODAF: The overall aims of an enterprise over a given period of time.

DoDAF: (DoDAF::Vision): An end that describes the future state of the enterprise, without regard to how it is to be achieved; a mental image of what the future will or could be like.

Base Classifier

- [Desirer](#)
- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EnterpriseVision[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$EnterpriseVision[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnterpriseVision[i].endBoundaryType</code>
enterprisePhase	EnterprisePhase	EnterpriseVision	<code>\$EnterpriseVision[i].enterprisePhase</code>
propertySet	PropertySet	UPDMElement	<code>\$EnterpriseVision[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnterpriseVision[i].startBoundaryType</code>
statement	VisionStatement	EnterpriseVision	<code>\$EnterpriseVision[i].statement</code>
URI	String	UPDMElement	<code>\$EnterpriseVision[i].URI</code>

269. Entity

A persistent information component representing a business concept.

270. EntityAttribute

MODAF: A defined property of an EntityItem.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EntityAttribute[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$EntityAttribute[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EntityAttribute[i].endBoundaryType</code>
EntityAttribute.canBeAppliedTo		EntityAttribute	<code>\$EntityAttribute[i].EntityAttribute.canBeAppliedTo</code>

propertySet	PropertySet	UPDMElement	<code>\$EntityAttribute[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EntityAttribute[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$EntityAttribute[i].URI</code>

271. EntityItem

MODAF: (MODAF::Entity): A definition (type) of an item of interest.

DoDAF: NA

Base Classifier

- [SubjectOfOperationalConstraint](#)
- [SubjectOfResourceConstraint](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EntityItem[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$EntityItem[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EntityItem[i].endBoundaryType</code>
EntityItem.ownedAttribute		EntityItem	<code>\$EntityItem[i].EntityItem.ownedAttribute</code>
propertySet	PropertySet	UPDMElement	<code>\$EntityItem[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EntityItem[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$EntityItem[i].URI</code>

272. EntityRelationship

MODAF: Asserts that there is a relationship between two EntityItems.

DoDAF: (DoDAF::DataAssociation): A relationship or association between two elements of proceduralized information.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EntityRelationship[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$EntityRelationship[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EntityRelationship[i].endBoundaryType</code>
EntityRelationship.endType		EntityRelationship	<code>\$EntityRelationship[i].EntityRelationship.endType</code>
propertySet	PropertySet	UPDMElement	<code>\$EntityRelationship[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EntityRelationship[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$EntityRelationship[i].URI</code>

273. Environment

MODAF: A definition of the conditions in which something exists or functions.

DoDAF: NA

Base Classifier

- [ConditionType](#)
- [PropertySet](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Environment[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Environment[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Environment[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Environment[i].endBoundaryType</code>
Environment.ownedAttributes		Environment	<code>\$Environment[i].Environment.ownedAttributes</code>
propertySet	PropertySet	UPDMElement	<code>\$Environment[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Environment[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Environment[i].URI</code>

274. Environmental effect

An Environmental effect is an influence on the system from the environment without communicating with it directly. For example Temperature or Humidity.

275. EnvironmentProperty

MODAF: Asserts that an Environment has one or more properties. These may be Climate, LocationType, or LightCondition.

DoDAF: NA

Base Classifier

- [Property](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$EnvironmentProperty[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$EnvironmentProperty[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$EnvironmentProperty[i].endBoundaryType</code>
EnvironmentalProperty.class		EnvironmentProperty	<code>\$EnvironmentProperty[i].EnvironmentalProperty.class</code>
EnvironmentalProperty.type		EnvironmentProperty	<code>\$EnvironmentProperty[i].EnvironmentalProperty.type</code>

maxValue	String	Property	\$EnvironmentProperty[i].maxValue
minValue	String	Property	\$EnvironmentProperty[i].minValue
propertySet	PropertySet	UPDMElement	\$EnvironmentProperty[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$EnvironmentProperty[i].startBoundaryType
URI	String	UPDMElement	\$EnvironmentProperty[i].URI

276. ER Diagram

Base Classifier

- [InvisibleStereotype](#)

277. errorIcon

Base Classifier

- [imaged](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
highlightColor	String	imaged	\$errorIcon[i].highlightColor

278. Essential

279. Exchange

UPDM: Abstract grouping for interactions that exchange messages.

MODAF:NA

DoDAF:NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Exchange[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Exchange[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Exchange[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$Exchange[i].propertySet

startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Exchange[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Exchange[i].URI</code>

280. ExchangeElement

MODAF: A relationship specifying the need to exchange information between nodes.

DoDAF: NA - this is a specialization of OperationalExchange (DoDAF::Interface).

Base Classifier

- [OperationalExchangeItem](#)
- [ResourceInteractionItem](#)
- [SubjectOfOperationalConstraint](#)
- [SubjectOfResourceConstraint](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$ExchangeElement[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ExchangeElement[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$ExchangeElement[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$ExchangeElement[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$ExchangeElement[i].conformsTo</code>
definedBy	EntityItem	ExchangeElement	<code>\$ExchangeElement[i].definedBy</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExchangeElement[i].endBoundaryType</code>
exchangeElementKind	ExchangeElementKind	ExchangeElement	<code>\$ExchangeElement[i].exchangeElementKind</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$ExchangeElement[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$ExchangeElement[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$ExchangeElement[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExchangeElement[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ExchangeElement[i].URI</code>

281. ExchangeElementKind

Enumeration of the types of element being exchanged on an information exchange.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
DataElement	Enumeration Literal	ExchangeElementKind	<code>\$ExchangeElementKind[i].DataElement</code>
InformationElement	Enumeration Literal	ExchangeElementKind	<code>\$ExchangeElementKind[i].InformationElement</code>

282. Executable

Denotes a program that may be run on a node.

Denotes a program file that can be executed on a computer system. Subclass of <<file>>.

Base Classifier

- [File](#)

283. Exhibits

UPDM: Relationship between a Node and a capability the node provides.

MODAF: (MODAF::CapabilityForNode): An assertion that a Node is required to have a Capability.

DoDAF: A couple that represents the capability that a performer manifests.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Exhibits[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Exhibits[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Exhibits[i].endBoundaryType
environmentalConditions	Environment	Exhibits	\$Exhibits[i].environmentalConditions
Exhibits.client		Exhibits	\$Exhibits[i].Exhibits.client
Exhibits.supplier		Exhibits	\$Exhibits[i].Exhibits.supplier
propertySet	PropertySet	UPDMElement	\$Exhibits[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$Exhibits[i].startBoundaryType
universalCapabilitySet	ActualPropertySet	Exhibits	\$Exhibits[i].universalCapabilitySet
URI	String	UPDMElement	\$Exhibits[i].URI

284. ExpansionKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
iterative	Enumeration Literal	ExpansionKind	\$ExpansionKind[i].iterative
parallel	Enumeration Literal	ExpansionKind	\$ExpansionKind[i].parallel
stream	Enumeration Literal	ExpansionKind	\$ExpansionKind[i].stream

285. Expose

286. Expose

A dependency between a service interface and a capability. The service interface exposes the capability.

287. extendedRequirement

A mix-in stereotype that contains generally useful attributes for requirements

Base Classifier

- [Requirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	<code>\$extendedRequirement[i].Derived</code>
DerivedFrom	Requirement	Requirement	<code>\$extendedRequirement[i].DerivedFrom</code>
Id	String	Requirement	<code>\$extendedRequirement[i].Id</code>
Master	Requirement	Requirement	<code>\$extendedRequirement[i].Master</code>
RefinedBy	NamedElement	Requirement	<code>\$extendedRequirement[i].RefinedBy</code>
risk	RiskKind	extendedRequirement	<code>\$extendedRequirement[i].risk</code>
SatisfiedBy	NamedElement	Requirement	<code>\$extendedRequirement[i].SatisfiedBy</code>
source	String	extendedRequirement	<code>\$extendedRequirement[i].source</code>
Text	String	Requirement	<code>\$extendedRequirement[i].Text</code>
TracedTo	NamedElement	Requirement	<code>\$extendedRequirement[i].TracedTo</code>
VerifiedBy	NamedElement	Requirement	<code>\$extendedRequirement[i].VerifiedBy</code>
verifyMethod	VerificationMethodKind	extendedRequirement	<code>\$extendedRequirement[i].verifyMethod</code>

288. External

An External block is a block that represents an actor. It facilitates a more detailed modeling of actors like ports or internal structure.

Base Classifier

- [Block](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	<code>\$External[i].isEncapsulated</code>

289. External system

An External system is a system that interacts with the system under development. For example an Information server or a Monitoring system.

290. ExternalIndividual

MODAF: An individual (i.e. something which has spatial and temporal extent) defined by an external ontology.

DoDAF: NA

Base Classifier

- [OntologyReference](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ExternalIndividual[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ExternalIndividual[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalIndividual[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ExternalIndividual[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalIndividual[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ExternalIndividual[i].URI</code>
url	String	OntologyReference	<code>\$ExternalIndividual[i].url</code>

291. ExternalTuple

UPDM: An instance of ExternalTupleType defined in an external Ontology.

MODAF:NA

DoDAF:NA

Base Classifier

- [OntologyReference](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ExternalTuple[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ExternalTuple[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalTuple[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ExternalTuple[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalTuple[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ExternalTuple[i].URI</code>
url	String	OntologyReference	<code>\$ExternalTuple[i].url</code>

292. ExternalTupleType

UPDM: An TupleType defined in an external Ontology.

MODAF:NA

DoDAF:NA

Base Classifier

- [ExternalType](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ExternalTupleType[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ExternalTupleType[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalTupleType[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ExternalTupleType[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalTupleType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ExternalTupleType[i].URI</code>
url	String	OntologyReference	<code>\$ExternalTupleType[i].url</code>

293. ExternalType

MODAF: A type defined by an external ontology.

DoDAF: NA

Base Classifier

- [OntologyReference](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ExternalType[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ExternalType[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalType[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ExternalType[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ExternalType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ExternalType[i].URI</code>
url	String	OntologyReference	<code>\$ExternalType[i].url</code>

294. fatallcon

Base Classifier

- [imaged](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
highlightColor	String	imaged	\$fatalIcon[i].highlightColor

295. FeatureDirection

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
provided	Enumeration Literal	FeatureDirection	\$FeatureDirection[i].provided
providedRequired	Enumeration Literal	FeatureDirection	\$FeatureDirection[i].providedRequired
required	Enumeration Literal	FeatureDirection	\$FeatureDirection[i].required

296. FieldedCapability

MODAF: An actual, fully-realised capability. A FieldedCapability must indicate its configuration CapabilityConfiguration.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$FieldedCapability[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$FieldedCapability[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$FieldedCapability[i].endBoundaryType
FieldedCapability.classifier		FieldedCapability	\$FieldedCapability[i].FieldedCapability.classifier
propertySet	PropertySet	UPDMElement	\$FieldedCapability[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$FieldedCapability[i].startBoundaryType
URI	String	UPDMElement	\$FieldedCapability[i].URI

297. FigureAlignKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
center	Enumeration Literal	FigureAlignKind	\$FigureAlignKind[i].center

justify	Enumeration Literal	FigureAlignKind	\$FigureAlignKind[i].justify
left	Enumeration Literal	FigureAlignKind	\$FigureAlignKind[i].left
right	Enumeration Literal	FigureAlignKind	\$FigureAlignKind[i].right

298. File

A physical file in the context of the system developed.

299. fileView

300. FillsPost

UPDM: Asserts that ActualPerson fills an ActualPost.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$FillsPost[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$FillsPost[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$FillsPost[i].endBoundaryType
endDate	ISO8601DateTime	FillsPost	\$FillsPost[i].endDate
FillsPost.client		FillsPost	\$FillsPost[i].FillsPost.client
FillsPost.supplier		FillsPost	\$FillsPost[i].FillsPost.supplier
propertySet	PropertySet	UPDMElement	\$FillsPost[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$FillsPost[i].startBoundaryType
startDate	ISO8601DateTime	FillsPost	\$FillsPost[i].startDate
URI	String	UPDMElement	\$FillsPost[i].URI

301. Findings

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
doctrine findings	Enumeration Literal	Findings	\$Findings[i].doctrine findings

education findings	Enumeration Literal	Findings	\$Findings[i].education findings
facility findings	Enumeration Literal	Findings	\$Findings[i].facility findings
leadership findings	Enumeration Literal	Findings	\$Findings[i].leadership findings
materiel findings	Enumeration Literal	Findings	\$Findings[i].materiel findings
organization findings	Enumeration Literal	Findings	\$Findings[i].organization findings
personnel findings	Enumeration Literal	Findings	\$Findings[i].personnel findings
training findings	Enumeration Literal	Findings	\$Findings[i].training findings
warfighter findings	Enumeration Literal	Findings	\$Findings[i].warfighter findings

302. FlowDirection

FlowDirection is an enumeration type that defines literals used for specifying input and output directions. *FlowDirection* is used by flow properties to indicate if a property is an input or an output with respect to its owner.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
in	Enumeration Literal	FlowDirection	\$FlowDirection[i].in
inout	Enumeration Literal	FlowDirection	\$FlowDirection[i].inout
out	Enumeration Literal	FlowDirection	\$FlowDirection[i].out

303. FlowPort

FlowPort is an interaction point through which input and/or output of items such as data, material, or energy may flow. This enables the owning block to declare which items it may exchange with its environment and the interaction points through which the exchange is made. We distinguish between atomic flow port and a nonatomic flow port. Atomic flow ports relay items that are classified by a single Block, ValueType, DataType, or Signal classifier. A nonatomic flow port relays items of several types as specified by a FlowSpecification. Flow ports and associated flow specifications define “what can flow” between the block and its environment, whereas item flows specify “what does flow” in a specific usage context. Flow ports relay items to their owning block or to a connector that connects them with their owner’s internal parts (internal connector).

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
direction	FlowDirection	FlowPort	\$FlowPort[i].direction
isAtomic	Boolean	FlowPort	\$FlowPort[i].isAtomic

304. FlowProperty

FlowProperty signifies a single flow element that can flow to/from a block. A flow property’s values are either received from or transmitted to an external block. Flow properties are defined directly on blocks or flow specifications that are those specifications which type the flow ports. Flow properties enable item flows across connectors connecting parts of the corresponding block types, either directly (in case of the property is defined on the block) or via flowPorts. For Block, Data Type, and Value Type properties, setting an “out” FlowProperty value of a block usage on one end of a connector will result in assigning the same value of an “in” FlowProperty of a block usage at the other end of the connector, provided the flow properties are matched. Flow properties of type Signal imply sending and/or receiving of a signal usage. An

“out” *FlowProperty* of type *Signal* means that the owning *Block* may broadcast the signal via connectors and an “in” *FlowProperty* means that the owning block is able to receive the *Signal*.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
direction	FlowDirection	FlowProperty	<code>\$FlowProperty[i].direction</code>

305. FlowSpecification

A *FlowSpecification* specifies inputs and outputs as a set of flow properties. A flow specification is used by flow ports to specify what items can flow via the port.

306. fmu

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
fileName	String	fmu	<code>\$fmu[i].fileName</code>
FMIVersion	String	fmu	<code>\$fmu[i].FMIVersion</code>
GUID	String	fmu	<code>\$fmu[i].GUID</code>
modelIdentifier	String	fmu	<code>\$fmu[i].modelIdentifier</code>
modelName	String	fmu	<code>\$fmu[i].modelName</code>

307. Focus

A class that defines the core logic or control flow for one or more auxiliary classes that support it. Support classes may be defined explicitly using Auxiliary classes or implicitly by dependency relationships. Focus classes are typically used together with one or more Auxiliary classes, and are particularly useful for specifying the core business logic or control flow of components during design. See also: «auxiliary».

308. Forecast

MODAF: A statement about the future state of one or more types of system or standard.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Forecast[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Forecast[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Forecast[i].endBoundaryType</code>
endDate	ISO8601DateTime	Forecast	<code>\$Forecast[i].endDate</code>
Forecast.client		Forecast	<code>\$Forecast[i].Forecast.client</code>
Forecast.pair		Forecast	<code>\$Forecast[i].Forecast.pair</code>
Forecast.supplier		Forecast	<code>\$Forecast[i].Forecast.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$Forecast[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Forecast[i].startBoundaryType</code>
startDate	ISO8601DateTime	Forecast	<code>\$Forecast[i].startDate</code>
URI	String	UPDMElement	<code>\$Forecast[i].URI</code>

309. Framework

A package that contains model elements which specify a reusable architecture for all or part of a system. Frameworks typically include classes, patterns or templates. When frameworks are specialized for an application domain, they are sometimes referred to as application frameworks.

310. FullPort

311. Function

MODAF: An activity which is specified in context of the resource (human or machine) that performs it.

DoDAF: Activity: Work, not specific to a single organization, weapon system or individual that transforms inputs (Resources) into outputs (Resources) or changes their state.

Base Classifier

- [Activity](#)
- [SubjectOfResourceConstraint](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
activityPerformableUnderConditio	Environment	Activity	<code>\$Function[i].activityPerformableUnderConditio</code>

n			on
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Function[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Function[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Function[i].endBoundaryType</code>
Function.ownedParameter		Function	<code>\$Function[i].Function.ownedParameter</code>
propertySet	PropertySet	UPDMElement	<code>\$Function[i].propertySet</code>
realizedBy	ResourceOperation	Function	<code>\$Function[i].realizedBy</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Function[i].startBoundaryType</code>
subject	ResourceInteractionItem	Function	<code>\$Function[i].subject</code>
URI	String	UPDMElement	<code>\$Function[i].URI</code>

312. FunctionAction

UPDM Artifact: The FunctionAction is defined as a call behavior action that invokes the function that needs to be performed. --This concept is required for mapping the architecture with UML and does not have a DoDAF or MoDAF equivalent.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$FunctionAction[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$FunctionAction[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$FunctionAction[i].endBoundaryType</code>
FunctionAction.activity		FunctionAction	<code>\$FunctionAction[i].FunctionAction.activity</code>
FunctionAction.behavior		FunctionAction	<code>\$FunctionAction[i].FunctionAction.behavior</code>
propertySet	PropertySet	UPDMElement	<code>\$FunctionAction[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$FunctionAction[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$FunctionAction[i].URI</code>

313. functionalRequirement

Requirement that specifies an operation or behavior that a system, or part of a system, must perform.

Base Classifier

- [extendedRequirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	<code>\$functionalRequirement[i].Derived</code>
DerivedFrom	Requirement	Requirement	<code>\$functionalRequirement[i].DerivedFrom</code>
Id	String	Requirement	<code>\$functionalRequirement[i].Id</code>
Master	Requirement	Requirement	<code>\$functionalRequirement[i].Master</code>

RefinedBy	NamedElement	Requirement	\$functionalRequirement[i].RefinedBy
risk	RiskKind	extendedRequirement	\$functionalRequirement[i].risk
SatisfiedBy	NamedElement	Requirement	\$functionalRequirement[i].SatisfiedBy
source	String	extendedRequirement	\$functionalRequirement[i].source
Text	String	Requirement	\$functionalRequirement[i].Text
TracedTo	NamedElement	Requirement	\$functionalRequirement[i].TracedTo
VerifiedBy	NamedElement	Requirement	\$functionalRequirement[i].VerifiedBy
verifyMethod	VerificationMethodKind	extendedRequirement	\$functionalRequirement[i].verifyMethod

314. FunctionalStandard

MODAF:NA

DoDAF:Functional standards set forth rules, conditions, guidelines, and characteristics.

Base Classifier

- Standard

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$FunctionalStandard[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$FunctionalStandard[i].conformsTo
currentStatus	String	Standard	\$FunctionalStandard[i].currentStatus
endBoundaryType	ISO8601DateTime	UPDMElement	\$FunctionalStandard[i].endBoundaryType
InformationTechnologyStandardCategory	String	Standard	\$FunctionalStandard[i].InformationTechnologyStandardCategory
mandatedDate	ISO8601DateTime	Standard	\$FunctionalStandard[i].mandatedDate
propertySet	PropertySet	UPDMElement	\$FunctionalStandard[i].propertySet
ratifiedBy	ActualOrganization	Standard	\$FunctionalStandard[i].ratifiedBy
retiredDate	ISO8601DateTime	Standard	\$FunctionalStandard[i].retiredDate
shortName	String	Standard	\$FunctionalStandard[i].shortName
startBoundaryType	ISO8601DateTime	UPDMElement	\$FunctionalStandard[i].startBoundaryType
URI	String	UPDMElement	\$FunctionalStandard[i].URI
version	String	Standard	\$FunctionalStandard[i].version

315. FunctionEdge

UPDM: An extension of <<ActivityEdge>> that is used to model the flow of control/objects through a Function.

MODAF: A FunctionEdge (MODAF::FunctionFlow) is a UML::ObjectFlow between Functions. NOTE: this has been extended in UPDM to additionally include UML::ControlFlows.

Base Classifier

- UPDMElement

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$FunctionEdge[i].actualPropertySet</code>
carriedItem	ResourceInteractionItem	FunctionEdge	<code>\$FunctionEdge[i].carriedItem</code>
conformsTo	Standard	UPDMElement	<code>\$FunctionEdge[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$FunctionEdge[i].endBoundaryType</code>
FunctionEdge.owner		FunctionEdge	<code>\$FunctionEdge[i].FunctionEdge.owner</code>
propertySet	PropertySet	UPDMElement	<code>\$FunctionEdge[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$FunctionEdge[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$FunctionEdge[i].URI</code>

316. GanttChartDiagram

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
entryData	String	GanttChartDiagram	<code>\$GanttChartDiagram[i].entryData</code>
hiddenEntryData	String	GanttChartDiagram	<code>\$GanttChartDiagram[i].hiddenEntryData</code>

317. GeoPoliticalExtent

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$GeoPoliticalExtent[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$GeoPoliticalExtent[i].conformsTo</code>
customKind	String	GeoPoliticalExtent	<code>\$GeoPoliticalExtent[i].customKind</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$GeoPoliticalExtent[i].endBoundaryType</code>
geoPoliticalExtentKind	GeoPoliticalExtentKind	GeoPoliticalExtent	<code>\$GeoPoliticalExtent[i].geoPoliticalExtentKind</code>
propertySet	PropertySet	UPDMElement	<code>\$GeoPoliticalExtent[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$GeoPoliticalExtent[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$GeoPoliticalExtent[i].URI</code>

318. GeoPoliticalExtentKind

Enumeration of geopolitical extent kinds, used to support the *geoPoliticalExtentKind* tag of the *geoPoliticalExtent* stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Country	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].Country</code>
Facility	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].Facility</code>
GeoFeature	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].GeoFeature</code>
Installation	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].Installation</code>
Other	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].Other</code>
RegionOfCountry	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].RegionOfCountry</code>
RegionOfWorld	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].RegionOfWorld</code>
Site	Enumeration Literal	GeoPoliticalExtentKind	<code>\$GeoPoliticalExtentKind[i].Site</code>

319. GeoPoliticalExtentType

MODAF:NA

DoDAF:A geospatial extent whose boundaries are by declaration or agreement by political parties.

Base Classifier

- [ConditionType](#)
- [OperationalExchangeItem](#)
- [ResourceInteractionItem](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$GeoPoliticalExtentType[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$GeoPoliticalExtentType[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$GeoPoliticalExtentType[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$GeoPoliticalExtentType[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$GeoPoliticalExtentType[i].conformsTo</code>
customKind	String	GeoPoliticalExtentType	<code>\$GeoPoliticalExtentType[i].customKind</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$GeoPoliticalExtentType[i].endBoundaryType</code>
geoPoliticalExtentTypeKind	GeoPoliticalExtentTypeKind	GeoPoliticalExtentType	<code>\$GeoPoliticalExtentType[i].geoPoliticalExtentTypeKind</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$GeoPoliticalExtentType[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$GeoPoliticalExtentType[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$GeoPoliticalExtentType[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$GeoPoliticalExtentType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$GeoPoliticalExtentType[i].URI</code>

320. GeoPoliticalExtentTypeKind

Enumeration of kinds of geopolitical extent type, derived from DoDAF, used to support the *geoPoliticalExtentTypeKind* tag of the *GeopoliticalExtentType* stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
CountryType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].CountryType</code>
FacilityType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].FacilityType</code>
GeoFeatureType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].GeoFeatureType</code>
InstallationType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].InstallationType</code>
OtherType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].OtherType</code>
RegionOfCountryType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].RegionOfCountryType</code>
RegionOfWorldType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].RegionOfWorldType</code>
SiteType	Enumeration Literal	GeoPoliticalExtentTypeKind	<code>\$GeoPoliticalExtentTypeKind[i].SiteType</code>

321. getter

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getter/setter for attribute	Element	getter	<code>\$getter[i].getter/setter for attribute</code>

322. GranularityLevel

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
global	Enumeration Literal	GranularityLevel	<code>\$GranularityLevel[i].global</code>
operational	Enumeration Literal	GranularityLevel	<code>\$GranularityLevel[i].operational</code>
strategic	Enumeration Literal	GranularityLevel	<code>\$GranularityLevel[i].strategic</code>
tactical	Enumeration Literal	GranularityLevel	<code>\$GranularityLevel[i].tactical</code>

323. grouped

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
group	CustomizationGroupNames	grouped	<code>\$grouped[i].group</code>

324. hasGroupName

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
groupName	String	hasGroupName	<code>\$hasGroupName[i].groupName</code>

325. HighLevelOperationalConcept

MODAF: A generalized model for operations.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$HighLevelOperationalConcept[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$HighLevelOperationalConcept[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$HighLevelOperationalConcept[i].endBoundaryType</code>
HighLevelOperationalConcept.ownedAttribute		HighLevelOperationalConcept	<code>\$HighLevelOperationalConcept[i].HighLevelOperationalConcept.ownedAttribute</code>
mission	Mission	HighLevelOperationalConcept	<code>\$HighLevelOperationalConcept[i].mission</code>
propertySet	PropertySet	UPDMElement	<code>\$HighLevelOperationalConcept[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$HighLevelOperationalConcept[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$HighLevelOperationalConcept[i].URI</code>

326. HyperlinkOwner

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hyperlinkModel	Element	HyperlinkOwner	<code>\$HyperlinkOwner[i].hyperlinkModel</code>

hyperlinkModelActive	Element	HyperlinkOwner	<code>\$HyperlinkOwner[i].hyperlinkModelActive</code>
hyperlinkText	String	HyperlinkOwner	<code>\$HyperlinkOwner[i].hyperlinkText</code>
hyperlinkTextActive	String	HyperlinkOwner	<code>\$HyperlinkOwner[i].hyperlinkTextActive</code>

327. iconHolder

328. IdentifiableElement

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
identifier	String	IdentifiableElement	<code>\$IdentifiableElement[i].identifier</code>

329. imaged

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
highlightColor	String	imaged	<code>\$imaged[i].highlightColor</code>

330. ImageFormat

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Joint Photographic Experts Group (*.jpg)	Enumeration Literal	ImageFormat	<code>\$ImageFormat[i].Joint Photographic Experts Group (*.jpg)</code>
Portable Network Graphics (*.png)	Enumeration Literal	ImageFormat	<code>\$ImageFormat[i].Portable Network Graphics (*.png)</code>
Scalable Vector Graphics (*.svg)	Enumeration Literal	ImageFormat	<code>\$ImageFormat[i].Scalable Vector Graphics (*.svg)</code>
Windows Enhanced Metafile (*.emf)	Enumeration Literal	ImageFormat	<code>\$ImageFormat[i].Windows Enhanced Metafile (*.emf)</code>
Windows Metafile (*.wmf)	Enumeration Literal	ImageFormat	<code>\$ImageFormat[i].Windows Metafile (*.wmf)</code>

331. Implement

A component definition that is not intended to have a specification itself. Rather, it is an implementation for a separate «specification» to which it has a Dependency.

332. Implementability

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
notional	Enumeration Literal	Implementability	\$Implementability[i].notional
real	Enumeration Literal	Implementability	\$Implementability[i].real

333. Implementation Map

The **Implementation Map** depicts the implementation relationships of a Performer (or MODAF Node). The predefined map includes the **Performer** (or MODAF Node) itself and the **System Resources** that have been indicated as implementing that operational element. For example, the Capability Configuration, Organization Type, or Software element that implements the Performer or Node. An Implementation Map also presents the internal compositions of each mapped element as well as the relationships among these elements.

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:


Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node  / .

Move the whole structure - click on the empty place in the Relation Map and drag.


Move the selected Node - click on the Node and drag.

Insert New Element - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.

Insert New Element of the Same Type - type the name and press Ctrl+Enter to create an element of the same type.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

334. Implementation Matrix

The Implementation Matrix describes the mapping between the Systems and Operational Elements.

The Rows of this matrix are Systems Elements (Resources, Functions and Resource Interactions) and the Columns are Operational Elements (Performers (DoDAF) / Nodes (MODAF and NAF), Operational Activities and Operational Exchanges).

Matrix can also be used to represent implementation of Service Functions by Functions from Systems viewpoint.

To build the Matrix:

- 1. Specify Rows** scope (Systems Elements (Resources, Functions and Resource Interactions), Service Functions);
- 2. Specify Columns** scope (Operational Elements (Performers (DoDAF) / Nodes (MODAF and NAF), Operational Activities and Operational Exchanges), Functions);
- 3. Click "Refresh"** button.

Systems Elements maps to Operational Elements by "Implements" relationship.

To map Systems Element to Operational Element, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

There are several types of predefined implementation matrices:

- *Performers Implementation Matrix - maps Performers to Resources (DoDAF only)*
- *Nodes Implementation Matrix - maps Nodes to Resources (MODAF and NAF only).*
- *Operational Activities Implementation Matrix - maps Operational Activities to Functions.*
- *Operational Exchanges Implementation Matrix - maps Operational Exchanges to Resource Interactions.*

335. ImplementationClass

The implementation of a class in some programming language (e.g., C++, Smalltalk, Java) in which an instance may not have more than one class. This is in contrast to Class, for which an instance may have multiple classes at one time and may gain or lose classes over time, and an object (a child of instance) may dynamically have multiple classes.

An Implementation class is said to realize a Classifier if it provides all of the operations defined for the Classifier with the same behavior as specified for the Classifier's operations. An Implementation Class may realize a number of different Types. Note that the physical attributes and associations of the Implementation class do not have to be the same as those of any Classifier it realizes and that the Implementation Class may provide methods for its operations in terms of its physical attributes and associations. See also: «type».

336. implementationModel

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
implementationModelID	int	implementationModel	\$implementationModel[i].implementationModelID

337. Implements

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Implements[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Implements[i].conformsTo

endBoundaryType	ISO8601DateTime	UPDMElement	\$Implements[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$Implements[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$Implements[i].startBoundaryType
URI	String	UPDMElement	\$Implements[i].URI

338. IncrementMilestone

MODAF: (MODAF::CapabilityIncrement): An ActualProjectMilestone (MODAF::ProjectMilestone) that indicates the point in time at which a project is predicted to deliver or has delivered a Capability.

DoDAF: NA

Base Classifier

- [ActualProjectMilestone](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualProjectMilestone.classifier		ActualProjectMilestone	\$IncrementMilestone[i].ActualProjectMilestone.classifier
ActualProjectMilestone.slot		ActualProjectMilestone	\$IncrementMilestone[i].ActualProjectMilestone.slot
actualPropertySet	ActualPropertySet	UPDMElement	\$IncrementMilestone[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$IncrementMilestone[i].conformsTo
date	ISO8601DateTime	ActualProjectMilestone	\$IncrementMilestone[i].date
description	String	ActualProjectMilestone	\$IncrementMilestone[i].description
endBoundaryType	ISO8601DateTime	UPDMElement	\$IncrementMilestone[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$IncrementMilestone[i].propertySet
resource	SystemResource	ActualProjectMilestone	\$IncrementMilestone[i].resource
startBoundaryType	ISO8601DateTime	UPDMElement	\$IncrementMilestone[i].startBoundaryType
URI	String	UPDMElement	\$IncrementMilestone[i].URI

339. IndividualPersonRole

UPDM: An individual person.

MODAF:NA

DoDAF: An Individual person.

Base Classifier

- [ActualPost](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualPost.classifier		ActualPost	\$IndividualPersonRole[i].ActualPost.classifi

			er
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$IndividualPersonRole[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$IndividualPersonRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$IndividualPersonRole[i].endBoundaryType</code>
filledBy	ActualPerson	ActualPost	<code>\$IndividualPersonRole[i].filledBy</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$IndividualPersonRole[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$IndividualPersonRole[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$IndividualPersonRole[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$IndividualPersonRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$IndividualPersonRole[i].URI</code>

340. Info

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
author	String	Info	<code>\$Info[i].author</code>
version	String	Info	<code>\$Info[i].version</code>

341. infoIcon

Base Classifier

- [imaged](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
highlightColor	String	imaged	<code>\$infoIcon[i].highlightColor</code>

342. Information

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Information[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Information[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Information[i].endBoundaryType</code>

informationKind	InformationKind	Information	<code>\$Information[i].informationKind</code>
propertySet	PropertySet	UPDMElement	<code>\$Information[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Information[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Information[i].URI</code>

343. InformationKind

Enumeration of kinds of information, derived from MODAF and DoDAF, used to support the InformationKind tag of the Information stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Data	Enumeration Literal	InformationKind	<code>\$InformationKind[i].Data</code>
DomainInformation	Enumeration Literal	InformationKind	<code>\$InformationKind[i].DomainInformation</code>
Information	Enumeration Literal	InformationKind	<code>\$InformationKind[i].Information</code>
PedigreeInformation	Enumeration Literal	InformationKind	<code>\$InformationKind[i].PedigreeInformation</code>
PositionReferenceFrame	Enumeration Literal	InformationKind	<code>\$InformationKind[i].PositionReferenceFrame</code>

344. InstanceTable

Base Classifier

- [DiagramTable](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
additionalElements	Element	DiagramTable	<code>\$InstanceTable[i].additionalElements</code>
autoResizeOn	Boolean	DiagramTable	<code>\$InstanceTable[i].autoResizeOn</code>
classifiers	Classifier	InstanceTable	<code>\$InstanceTable[i].classifiers</code>
columnElements	String	DiagramTable	<code>\$InstanceTable[i].columnElements</code>
columnIds	String	DiagramTable	<code>\$InstanceTable[i].columnIds</code>
columnWidth	Integer	DiagramTable	<code>\$InstanceTable[i].columnWidth</code>
customColumns	String	DiagramTable	<code>\$InstanceTable[i].customColumns</code>
defaultRowElementsOwner	Element	DiagramTable	<code>\$InstanceTable[i].defaultRowElementsOwner</code>
excludedElements	Element	DiagramTable	<code>\$InstanceTable[i].excludedElements</code>
hideColumns	String	DiagramTable	<code>\$InstanceTable[i].hideColumns</code>
query	StructuredExpression	DiagramTable	<code>\$InstanceTable[i].query</code>
rowElements	String	DiagramTable	<code>\$InstanceTable[i].rowElements</code>
rowElementType	Element	DiagramTable	<code>\$InstanceTable[i].rowElementType</code>
rowFilters	String	DiagramTable	<code>\$InstanceTable[i].rowFilters</code>
rowHeight	Integer	DiagramTable	<code>\$InstanceTable[i].rowHeight</code>
scope	Element	DiagramTable	<code>\$InstanceTable[i].scope</code>

showDetailedColumnName	Boolean	DiagramTable	<code>\$InstanceTable[i].showDetailedColumnName</code>
showFullPath	Boolean	DiagramTable	<code>\$InstanceTable[i].showFullPath</code>
sort	String	DiagramTable	<code>\$InstanceTable[i].sort</code>
typesIncludeSubtypes	boolean	DiagramTable	<code>\$InstanceTable[i].typesIncludeSubtypes</code>

345. Instantiate

A usage dependency among classifiers indicating that operations on the client create instances of the supplier.

346. Integer

Base Classifier

- [Number](#)

347. InteractionOperatorKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
alt	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].alt</code>
assert	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].assert</code>
break	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].break</code>
consider	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].consider</code>
critical	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].critical</code>
ignore	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].ignore</code>
loop	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].loop</code>
neg	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].neg</code>
opt	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].opt</code>
par	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].par</code>
seq	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].seq</code>
strict	Enumeration Literal	InteractionOperatorKind	<code>\$InteractionOperatorKind[i].strict</code>

348. InterfaceBlock

Base Classifier

- [Block](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	<code>\$InterfaceBlock[i].isEncapsulated</code>

349. interfaceRequirement

Requirement that specifies the ports for connecting systems and system parts and the optionally may include the item flows across the connector and/or Interface constraints.

Base Classifier

- [extendedRequirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	<code>\$interfaceRequirement[i].Derived</code>
DerivedFrom	Requirement	Requirement	<code>\$interfaceRequirement[i].DerivedFrom</code>
Id	String	Requirement	<code>\$interfaceRequirement[i].Id</code>
Master	Requirement	Requirement	<code>\$interfaceRequirement[i].Master</code>
RefinedBy	NamedElement	Requirement	<code>\$interfaceRequirement[i].RefinedBy</code>
risk	RiskKind	extendedRequirement	<code>\$interfaceRequirement[i].risk</code>
SatisfiedBy	NamedElement	Requirement	<code>\$interfaceRequirement[i].SatisfiedBy</code>
source	String	extendedRequirement	<code>\$interfaceRequirement[i].source</code>
Text	String	Requirement	<code>\$interfaceRequirement[i].Text</code>
TracedTo	NamedElement	Requirement	<code>\$interfaceRequirement[i].TracedTo</code>
VerifiedBy	NamedElement	Requirement	<code>\$interfaceRequirement[i].VerifiedBy</code>
verifyMethod	VerificationMethodKind	extendedRequirement	<code>\$interfaceRequirement[i].verifyMethod</code>

350. Interval

Interval distribution - unknown probability between min and max

Base Classifier

- [BasicInterval](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
max	Real	BasicInterval	<code>\$Interval[i].max</code>
min	Real	BasicInterval	<code>\$Interval[i].min</code>

351. invariant

352. InvisibleStereotype

353. InvocationOnNestedPortAction

Base Classifier

- [ElementPropertyPath](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
onNestedPort	Port	InvocationOnNestedPortAction	<code>\$InvocationOnNestedPortAction[i].onNestedPort</code>
propertyPath	Property	ElementPropertyPath	<code>\$InvocationOnNestedPortAction[i].propertyPath</code>

354. IsCapableOfPerforming

UPDM: Links a Performer to the behavior that it can perform.

DoDAF: The Performs (DoDAF::activityPerformedByPerformer) relationship is an overlap between a Performer and a PerformedActivity (DoDAF::Activity) wherein the activity is performed by the Performer.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$IsCapableOfPerforming[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$IsCapableOfPerforming[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$IsCapableOfPerforming[i].endBoundaryType</code>
Performs.client		IsCapableOfPerforming	<code>\$IsCapableOfPerforming[i].Performs.client</code>
Performs.supplier		IsCapableOfPerforming	<code>\$IsCapableOfPerforming[i].Performs.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$IsCapableOfPerforming[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$IsCapableOfPerforming[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$IsCapableOfPerforming[i].URI</code>

355. ISO8601DateTime

MODAF: A date and time specified in the ISO8601 date-time format including timezone designator (TZD): YYYY-MM-DDThh:mm:ssTZD.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ISO8601DateTime[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ISO8601DateTime[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ISO8601DateTime[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ISO8601DateTime[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ISO8601DateTime[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ISO8601DateTime[i].URI</code>

356. ItemFlow

An ItemFlow describes the flow of items across a connector or an association. It may constrain the item exchange between blocks, block usages, or flow ports as specified by their flow properties. For example, a pump connected to a tank: the pump has an “out” flow property of type Liquid and the tank has an “in” FlowProperty of type Liquid. To signify that only water flows between the pump and the tank, we can specify an ItemFlow of type Water on the connector.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
itemProperty	Property	ItemFlow	<code>\$ItemFlow[i].itemProperty</code>

357. JointPotentialDesignator

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
joint	Enumeration Literal	JointPotentialDesignator	<code>\$JointPotentialDesignator[i].joint</code>
not joint	Enumeration Literal	JointPotentialDesignator	<code>\$JointPotentialDesignator[i].not joint</code>

358. KnownResource

MODAF: Asserts that a known Resource plays a part in the architecture.

DoDAF: NA – covered by the more general temporalWholePart element.

Base Classifier

- [NodeRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$KnownResource[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$KnownResource[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$KnownResource[i].endBoundaryType</code>
KnownResrouce.type		KnownResource	<code>\$KnownResource[i].KnownResrouce.type</code>
NodeRole.class		NodeRole	<code>\$KnownResource[i].NodeRole.class</code>
NodeRole.type		NodeRole	<code>\$KnownResource[i].NodeRole.type</code>
performsInContext	OperationalActivity	NodeRole	<code>\$KnownResource[i].performsInContext</code>
propertySet	PropertySet	UPDMElement	<code>\$KnownResource[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$KnownResource[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$KnownResource[i].URI</code>

359. L1**Base Classifier**

- [InvisibleStereotype](#)

360. L1 Node Types**Base Classifier**

- [InvisibleStereotype](#)

361. L1 Package**Base Classifier**

- [InvisibleStereotype](#)

362. L1i**Base Classifier**

- [InvisibleStereotype](#)

- [OV-2i](#)

363. L2 Package

Base Classifier

- [InvisibleStereotype](#)

364. L2i

Base Classifier

- [InvisibleStereotype](#)
- [OV-2i](#)

365. L3

Node Interactions (L3) addresses operational exchanges between nodes. To fill in L3 table, Existing Operational Exchanges of any available kind such as Information Exchange, Materiel Exchange, Organizational Exchange, Energy Exchange and etc. have to be added to it.

*Click **Add New** button to create a new Operational Exchange between selected Nodes.*

*Click **Add Existing** button to select Operational Exchanges or Needlines. In case Needline is selected, all Operational Exchanges flowing via it will be added to the table.*

"Operational Exchange Identifier", "Operational Exchange Item Name, Producing and Consuming Operational Activities, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Operational Exchanges L1 product is recommended to use.

Rows (Operational Exchanges) can be removed from model or only from table, can be filtered according to represented Operational Exchange kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	L3	\$L3[i].hideColumns

366. L3 Package

Base Classifier

- [InvisibleStereotype](#)

367. L3 Report

Base Classifier

- [InvisibleStereotype](#)

368. L3 Role Based

The Node Role Interactions (L3) addresses the resources exchanged between node roles and the relevant attributes of the exchanges

To fill in L3 table, Existing Operational Exchanges of any available kind such as Information Exchange, Materiel Exchange, Organizational Exchange, Energy Exchange, Geo Political Extent Exchange and etc. have to be added to it.

*Click **Add Existing** button to select Operational Exchanges, Needlines, or Service Channels. In case Needline or Service Channel is selected, all Operational Exchanges flowing via it will be added to the table.*

"Operational Exchange ID", "Operational Exchange Item", "Producing and Consuming Operational Activities", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Operational Exchanges L2 diagram is recommended to use.

Rows (Operational Exchanges) can be removed from model or only from table, can be filtered according to represented Operational Exchange kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

hideColumns	String	L3 Role Based	\$L3RoleBased[i].hideColumns
-------------	--------	---------------	------------------------------

369. L3 Role Based Report

Base Classifier

- InvisibleStereotype

370. L4

Base Classifier

- InvisibleStereotype

371. L4 Package

Base Classifier

- InvisibleStereotype

372. L4-P4

The Systems Function to Operational Activity Traceability Matrix (L4-P4) addresses the linkage between Functions described in P4 and Operational Activities specified in L4. The Rows of this matrix are Functions and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Functions);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Functions implements Operational Activities using "Implements" relationship.

To map Function to Operational Activity, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

373. L5**Base Classifier**

- [InvisibleStereotype](#)

374. L5 Package**Base Classifier**

- [InvisibleStereotype](#)

375. L6**Base Classifier**

- [InvisibleStereotype](#)

376. L6 Package**Base Classifier**

- [InvisibleStereotype](#)

377. L7**Base Classifier**

- [InvisibleStereotype](#)

378. L7 Package**Base Classifier**

- [InvisibleStereotype](#)

379. L8

The Logical Constraints (P8) specifies operational or business rules that are constraints on an enterprise, a mission, operation, business, or architecture. There are two ways to fill this table.

- 1. Add new Operational Constraint. Click **Add New** button and select constrained Node, Operational Activity, Entity Item, Mission, Exchange Element, or Operational Exchange). Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. Add Existing Operational Constraints. Click **Add Existing** button and select Operational Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Operational Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	L8	<code>\$L8[i].hideColumns</code>

380. L8 Package

Base Classifier

- [InvisibleStereotype](#)

381. L8 Report

Base Classifier

- [InvisibleStereotype](#)

382. LegendItem

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
color	String	LegendItem	<code>\$LegendItem[i].color</code>
colorUse	boolean	LegendItem	<code>\$LegendItem[i].colorUse</code>
font	String	LegendItem	<code>\$LegendItem[i].font</code>
fontUse	boolean	LegendItem	<code>\$LegendItem[i].fontUse</code>
lineWidth	String	LegendItem	<code>\$LegendItem[i].lineWidth</code>
lineWidthUse	boolean	LegendItem	<code>\$LegendItem[i].lineWidthUse</code>
name	String	LegendItem	<code>\$LegendItem[i].name</code>
penColor	String	LegendItem	<code>\$LegendItem[i].penColor</code>
penColorUse	boolean	LegendItem	<code>\$LegendItem[i].penColorUse</code>
shape	boolean	LegendItem	<code>\$LegendItem[i].shape</code>
textColor	String	LegendItem	<code>\$LegendItem[i].textColor</code>
textColorUse	boolean	LegendItem	<code>\$LegendItem[i].textColorUse</code>
useFillColor	boolean	LegendItem	<code>\$LegendItem[i].useFillColor</code>
useFillColorUse	boolean	LegendItem	<code>\$LegendItem[i].useFillColorUse</code>

383. LegendLocation

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Above Column Header	Enumeration Literal	LegendLocation	<code>\$LegendLocation[i].Above Column Header</code>
Do Not Display	Enumeration Literal	LegendLocation	<code>\$LegendLocation[i].Do Not Display</code>
Top-Left Corner	Enumeration Literal	LegendLocation	<code>\$LegendLocation[i].Top-Left Corner</code>

384. Library

Denotes a static or dynamic library.

Denotes a static or dynamic library file. Subclass of <<file>>.

Base Classifier

- [File](#)

385. LightCondition

MODAF: a specification of environmental lighting conditions.

Base Classifier

- [Environment](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$LightCondition[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$LightCondition[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$LightCondition[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LightCondition[i].endBoundaryType</code>
Environment.ownedAttributes		Environment	<code>\$LightCondition[i].Environment.ownedAttributes</code>
propertySet	PropertySet	UPDMElement	<code>\$LightCondition[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LightCondition[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$LightCondition[i].URI</code>

386. Location

DoDAF: All subtypes of << IndividualType >> Location, such as Facility, Site, etc.

Base Classifier

- [ActualLocation](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Location[i].actualPropertySet</code>
address	String	ActualLocation	<code>\$Location[i].address</code>
conformsTo	Standard	UPDMElement	<code>\$Location[i].conformsTo</code>
customKind	String	ActualLocation	<code>\$Location[i].customKind</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Location[i].endBoundaryType</code>
locationKind	LocationKind	ActualLocation	<code>\$Location[i].locationKind</code>
locationNamedByAddress	Boolean	ActualLocation	<code>\$Location[i].locationNamedByAddress</code>
propertySet	PropertySet	UPDMElement	<code>\$Location[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Location[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Location[i].URI</code>

387. LocationHolder

UPDM: Abstract grouping to capture elements that can have a location.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$LocationHolder[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$LocationHolder[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LocationHolder[i].endBoundaryType</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$LocationHolder[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$LocationHolder[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$LocationHolder[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LocationHolder[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$LocationHolder[i].URI</code>

388. LocationKind

Enumeration of location kinds, used to support the locationKind tag of the LocationKind stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
CircularArea	Enumeration Literal	LocationKind	<code>\$LocationKind[i].CircularArea</code>
EllipticalArea	Enumeration Literal	LocationKind	<code>\$LocationKind[i].EllipticalArea</code>
GeoStationaryPoint	Enumeration Literal	LocationKind	<code>\$LocationKind[i].GeoStationaryPoint</code>
Line	Enumeration Literal	LocationKind	<code>\$LocationKind[i].Line</code>
Other	Enumeration Literal	LocationKind	<code>\$LocationKind[i].Other</code>
PlanarSurface	Enumeration Literal	LocationKind	<code>\$LocationKind[i].PlanarSurface</code>
Point	Enumeration Literal	LocationKind	<code>\$LocationKind[i].Point</code>
PolygonArea	Enumeration Literal	LocationKind	<code>\$LocationKind[i].PolygonArea</code>
RectangularArea	Enumeration Literal	LocationKind	<code>\$LocationKind[i].RectangularArea</code>
SolidVolume	Enumeration Literal	LocationKind	<code>\$LocationKind[i].SolidVolume</code>
Surface	Enumeration Literal	LocationKind	<code>\$LocationKind[i].Surface</code>

389. LocationType

MODAF: A general specification of the surroundings / scenario in which an operation may take place. Examples would be: "desert", "arctic", "at sea", etc.

DoDAF: A point or extent in space that may be referred to physically or logically. Includes concepts such as: Facility, Installation, RealProperty, Site, , and instances of conditions such as underwater (as specified in UJTLs).

Base Classifier

- [ConceptItem](#)
- [ConditionType](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$LocationType[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$LocationType[i].conformsTo</code>
customKind	String	LocationType	<code>\$LocationType[i].customKind</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LocationType[i].endBoundaryType</code>
locationTypeKind	LocationTypeKind	LocationType	<code>\$LocationType[i].locationTypeKind</code>
propertySet	PropertySet	UPDMElement	<code>\$LocationType[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LocationType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$LocationType[i].URI</code>

390. LocationTypeConceptRole**Base Classifier**

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$LocationTypeConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$LocationTypeConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$LocationTypeConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LocationTypeConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$LocationTypeConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LocationTypeConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$LocationTypeConceptRole[i].URI</code>

391. LocationTypeKind

Enumeration of kinds of location types, derived from DoDAF, used to support the LocationTypeKind tag of the LocationTypeKind stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
CircularAreaType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].CircularAreaType</code>
EllipticalAreaType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].EllipticalAreaType</code>
GeoStationaryPointType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].GeoStationaryPointType</code>

LineType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].LineType</code>
OtherType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].OtherType</code>
PlanarSurfaceType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].PlanarSurfaceType</code>
PointType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].PointType</code>
PolygonAreaType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].PolygonAreaType</code>
RectangularAreaType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].RectangularAreaType</code>
SolidVolumeType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].SolidVolumeType</code>
SurfaceType	Enumeration Literal	LocationTypeKind	<code>\$LocationTypeKind[i].SurfaceType</code>

392. Logical Viewpoint

Base Classifier

- [InvisibleStereotype](#)

393. LogicalArchitecture

MODAF: A CompositeStructureModel whose parts are either NodeRoles (MODAF::Node), ProblemDomains, or KnownResources.

DoDAF: NA

Base Classifier

- [NodeParent](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$LogicalArchitecture[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$LogicalArchitecture[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$LogicalArchitecture[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$LogicalArchitecture[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LogicalArchitecture[i].endBoundaryType</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$LogicalArchitecture[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$LogicalArchitecture[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$LogicalArchitecture[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LogicalArchitecture[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$LogicalArchitecture[i].URI</code>

394. LogicalDataModel

MODAF: A LogicalDataModel is a specification of business information requirements as a formal data structure, where relationships and classes (entities) are used to specify the logic which underpins the information.

DoDAF: A Logical Data Model allows analysis of an architecture's data definition aspect, without consideration of implementation specific or product specific issues.

Base Classifier

- [DataModel](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$LogicalDataModel[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$LogicalDataModel[i].conformsTo</code>
DataModel.ownedElement		DataModel	<code>\$LogicalDataModel[i].DataModel.ownedElement</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LogicalDataModel[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$LogicalDataModel[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$LogicalDataModel[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$LogicalDataModel[i].URI</code>

395. Lr

Base Classifier

- [InvisibleStereotype](#)

396. Lr Package

Base Classifier

- [InvisibleStereotype](#)

397. Lr Programme to Capability Mapping

Programme to Capability Mapping (Lr) describes the mapping of programs and projects to capabilities to show how the specific projects and program elements help to achieve a capability. The Rows of this matrix are Capabilities and the Columns are Projects.

To build the Matrix:

1. *Specify Rows scope (Capabilities);*
2. *Specify Columns scope (Projects);*
3. *Click "Refresh" button.*

There are two possible ways to map Capability and a Project:

1. Activity is part of Project ("Activity Part of Project" relationship and a part of Capability ("Activity Part of Capability" relationship).
2. Project owns Increment Milestones (Owned Milestones property) that are related to Resources (Resources property) exhibiting (Capability of Performer relationship) capabilities.

Both transitive structures of relations map Capabilities to Projects. The mapping is displayed in the Matrix.

Base Classifier

- [InvisibleStereotype](#)

398. Lrc

Base Classifier

- [InvisibleStereotype](#)

399. MapsToCapability

MODAF: Asserts that a StandardOperationalActivity is in some way part of a capability.

DoDAF: MapsToCapability (DoDAF::ActivityPartOfCapability) is a disposition to manifest an Activity. An Activity to be performed to achieve a desired effect under specified [performance] standards and conditions through combinations of ways and means.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$MapsToCapability[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$MapsToCapability[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$MapsToCapability[i].endBoundaryType</code>
MapsToCapability.client		MapsToCapability	<code>\$MapsToCapability[i].MapsToCapability.client</code>
MapsToCapability.supplier		MapsToCapability	<code>\$MapsToCapability[i].MapsToCapability.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$MapsToCapability[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$MapsToCapability[i].startBoundaryType</code>

URI	String	UPDMElement	\$MapsToCapability[i].URI
-----	--------	-----------------------------	---------------------------

400. Materiel

MODAF: Artifact, A type of man-made object. Examples are "car", "radio", "diesel", etc.

DoDAF: Equipment, apparatus or supplies that are of interest, without distinction as to its application for administrative or combat purposes.

Base Classifier

- [ResourceInteractionItem](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Materiel[i].actualPropertySet
affectedFunctions	Function	ResourceInteractionItem	\$Materiel[i].affectedFunctions
appliesTo	UPDMElement	PropertySet	\$Materiel[i].appliesTo
conformsTo	Standard	UPDMElement	\$Materiel[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Materiel[i].endBoundaryType
physicalLocation	ActualLocation	LocationHolder	\$Materiel[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$Materiel[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$Materiel[i].requiredEnvironment
startBoundaryType	ISO8601DateTime	UPDMElement	\$Materiel[i].startBoundaryType
URI	String	UPDMElement	\$Materiel[i].URI

401. MatrixFilter

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
columnElementType	Element	MatrixFilter	\$MatrixFilter[i].columnElementType
columnPropertyFilter	String	MatrixFilter	\$MatrixFilter[i].columnPropertyFilter
columnScope	Element	MatrixFilter	\$MatrixFilter[i].columnScope
columnScopeDefined	boolean	MatrixFilter	\$MatrixFilter[i].columnScopeDefined
columnTypesIncludeSubtypes	boolean	MatrixFilter	\$MatrixFilter[i].columnTypesIncludeSubtypes
removedColumnElements	Element	MatrixFilter	\$MatrixFilter[i].removedColumnElements
removedRowElements	Element	MatrixFilter	\$MatrixFilter[i].removedRowElements
rowElementType	Element	MatrixFilter	\$MatrixFilter[i].rowElementType
rowPropertyFilter	String	MatrixFilter	\$MatrixFilter[i].rowPropertyFilter
rowScope	Element	MatrixFilter	\$MatrixFilter[i].rowScope
rowScopeDefined	boolean	MatrixFilter	\$MatrixFilter[i].rowScopeDefined

rowTypesIncludeSubtypes	boolean	MatrixFilter	\$MatrixFilter[i].rowTypesIncludeSubtypes
-------------------------	---------	--------------	---

402. Measure

MODAF:NA

DoDAF: The magnitude of some attribute of an individual.

Base Classifier

- ActualPropertySet

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Measure[i].actualPropertySet
ActualPropertySet.classifier		ActualPropertySet	\$Measure[i].ActualPropertySet.classifier
ActualPropertySet.slot		ActualPropertySet	\$Measure[i].ActualPropertySet.slot
appliesTo	UPDMElement	ActualPropertySet	\$Measure[i].appliesTo
conformsTo	Standard	UPDMElement	\$Measure[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Measure[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$Measure[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$Measure[i].startBoundaryType
URI	String	UPDMElement	\$Measure[i].URI

403. Measurement

MODAF: MeasurableProperty: A property of something in the physical world, expressed in amounts of a unit of measure. The property may have a required value - either specified by the [defaultValue] from UML::property attribute, or the [minValue] and [maxValue] to specify a required range.

DoDAF: Measure: A Measurement (DoDAF::Measure) is the magnitude of some attribute of an individual.

Base Classifier

- Property

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Measurement[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Measurement[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Measurement[i].endBoundaryType
maxValue	String	Property	\$Measurement[i].maxValue
minValue	String	Property	\$Measurement[i].minValue
propertySet	PropertySet	UPDMElement	\$Measurement[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$Measurement[i].startBoundaryType
URI	String	UPDMElement	\$Measurement[i].URI

404. MeasurementSet

UPDM: A set or collection of Measurement(s).

MODAF: NA

DoDAF: NA

Base Classifier

- [PropertySet](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$MeasurementSet[i].actualPropertySet
appliesTo	UPDMElement	PropertySet	\$MeasurementSet[i].appliesTo
conformsTo	Standard	UPDMElement	\$MeasurementSet[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$MeasurementSet[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$MeasurementSet[i].propertySet
PropertySet.ownedAttribute		MeasurementSet	\$MeasurementSet[i].PropertySet.ownedAttribute
startBoundaryType	ISO8601DateTime	UPDMElement	\$MeasurementSet[i].startBoundaryType
URI	String	UPDMElement	\$MeasurementSet[i].URI

405. MeasureType

MODAF: NA

DoDAF: A category of Measures.

Base Classifier

- [MeasurementSet](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$MeasureType[i].actualPropertySet
appliesTo	UPDMElement	PropertySet	\$MeasureType[i].appliesTo
conformsTo	Standard	UPDMElement	\$MeasureType[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$MeasureType[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$MeasureType[i].propertySet
PropertySet.ownedAttribute		MeasurementSet	\$MeasureType[i].PropertySet.ownedAttribute
startBoundaryType	ISO8601DateTime	UPDMElement	\$MeasureType[i].startBoundaryType
URI	String	UPDMElement	\$MeasureType[i].URI

406. mergedDiagram

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
mergedFromBothContributors	Boolean	mergedDiagram	<code>\$mergedDiagram[i].mergedFromBothContributors</code>
mergedSymbols	String	mergedDiagram	<code>\$mergedDiagram[i].mergedSymbols</code>

407. MessageKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
complete	Enumeration Literal	MessageKind	<code>\$MessageKind[i].complete</code>
found	Enumeration Literal	MessageKind	<code>\$MessageKind[i].found</code>
lost	Enumeration Literal	MessageKind	<code>\$MessageKind[i].lost</code>
unknown	Enumeration Literal	MessageKind	<code>\$MessageKind[i].unknown</code>

408. MessageSort

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
asynchCall	Enumeration Literal	MessageSort	<code>\$MessageSort[i].asynchCall</code>
asynchSignal	Enumeration Literal	MessageSort	<code>\$MessageSort[i].asynchSignal</code>
createMessage	Enumeration Literal	MessageSort	<code>\$MessageSort[i].createMessage</code>
deleteMessage	Enumeration Literal	MessageSort	<code>\$MessageSort[i].deleteMessage</code>
reply	Enumeration Literal	MessageSort	<code>\$MessageSort[i].reply</code>
synchCall	Enumeration Literal	MessageSort	<code>\$MessageSort[i].synchCall</code>

409. MessageType

The specification of information exchanged between service consumers and providers.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
encoding	String	MessageType	<code>\$MessageType[i].encoding</code>

410. Metaclass

411. Metadata

MODAF: Annotation that can be applied to any element in the architecture.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Metadata[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Metadata[i].conformsTo</code>
dublinCoreElement	String	Metadata	<code>\$Metadata[i].dublinCoreElement</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Metadata[i].endBoundaryType</code>
modMetaDataElement	String	Metadata	<code>\$Metadata[i].modMetaDataElement</code>
name	String	Metadata	<code>\$Metadata[i].name</code>
propertySet	PropertySet	UPDMElement	<code>\$Metadata[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Metadata[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Metadata[i].URI</code>

412. MetaInfo

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Author		MetaInfo	<code>\$MetaInfo[i].Author</code>
Copyright Date	String	MetaInfo	<code>\$MetaInfo[i].Copyright Date</code>
Document Title	String	MetaInfo	<code>\$MetaInfo[i].Document Title</code>
Issue ID	String	MetaInfo	<code>\$MetaInfo[i].Issue ID</code>
Publishing Date	String	MetaInfo	<code>\$MetaInfo[i].Publishing Date</code>

413. Metamodel

A model of a model, that typically contains metaclasses.

414. metaProperty

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
newName	String	metaProperty	<code>\$metaProperty[i].newName</code>
newTypes	Type	metaProperty	<code>\$metaProperty[i].newTypes</code>
suggestedValues	String	metaProperty	<code>\$metaProperty[i].suggestedValues</code>

415. MetricDefinition

Metric property. Describes expression to calculate metric value

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
abbreviation	String	MetricDefinition	<code>\$MetricDefinition[i].abbreviation</code>

416. MetricInstance

417. MetricSuite

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
target	Element	MetricSuite	<code>\$MetricSuite[i].target</code>

418. migrationLog

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
logEntry	String	migrationLog	<code>\$migrationLog[i].logEntry</code>

419. Milestone

A Milestone is a means for depicting progress in behaviors in order to analyze liveness. Milestones are particularly useful for behaviors that are long lasting or even infinite.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
progress	Integer	Milestone	\$Milestone[i].progress
signal	Signal	Milestone	\$Milestone[i].signal
value	ValueSpecification	Milestone	\$Milestone[i].value

420. MilestoneSequence

MODAF: A MilestoneSequence (MODAF::MilestoneRelationship) is a relationship between two milestones.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$MilestoneSequence[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$MilestoneSequence[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$MilestoneSequence[i].endBoundaryType
MilestoneSequence.client		MilestoneSequence	\$MilestoneSequence[i].MilestoneSequence.client
MilestoneSequence.supplier		MilestoneSequence	\$MilestoneSequence[i].MilestoneSequence.supplier
propertySet	PropertySet	UPDMElement	\$MilestoneSequence[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$MilestoneSequence[i].startBoundaryType
URI	String	UPDMElement	\$MilestoneSequence[i].URI

421. Mission

MODAF: A purpose to which a person, organization or autonomous system is tasked.

DoDAF: The task, together with the purpose, that clearly indicates the action to be taken.

Base Classifier

- [SubjectOfOperationalConstraint](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Mission[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Mission[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Mission[i].endBoundaryType
missionArea	String	Mission	\$Mission[i].missionArea

propertySet	PropertySet	UPDMElement	\$Mission[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$Mission[i].startBoundaryType
URI	String	UPDMElement	\$Mission[i].URI

422. ModelLibrary

A package that contains model elements which are intended to be reused by other packages. Model libraries are frequently used in conjunction with applied profiles. This is expressed by defining a dependency between a profile and a model library package, or by defining a model library as contained in a profile package. The classes in a model library are not stereotypes and tagged definitions extending the metamodel. A model library is analogous to a class library in some programming languages.

When a model library is defined as a part of a profile, it is imported or deleted with the application or removal of the profile. The profile is implicitly applied to its model library. In the other case, when the model library is defined as an external package imported by a profile, the profile requires that the model library be there in the model at the stage of the profile application. The application or the removal of the profile does not affect the presence of the model library elements.

423. moe

A measure of effectiveness (moe) represents a parameter whose value is critical for achieving the desired mission cost effectiveness.

424. mount

425. NAF 4.0 All Views Report

Base Classifier

- [InvisibleStereotype](#)

426. NATO All View Viewpoint

Base Classifier

- [InvisibleStereotype](#)

427. NATO All Views Report

Base Classifier

- [InvisibleStereotype](#)

428. NATO Capability Viewpoint

Base Classifier

- [InvisibleStereotype](#)

429. NATO Operational Viewpoint

Base Classifier

- [InvisibleStereotype](#)

430. NATO Programme Viewpoint

Base Classifier

- [InvisibleStereotype](#)

431. NATO Service-Oriented Viewpoint

Base Classifier

- [InvisibleStereotype](#)

432. NATO Systems Viewpoint

Base Classifier

- [InvisibleStereotype](#)

433. NATO Technical Viewpoint

Base Classifier

- [InvisibleStereotype](#)

434. NAV-1

Base Classifier

- [InvisibleStereotype](#)

435. NAV-1 Package

Base Classifier

- [InvisibleStereotype](#)

436. NAV-1 Report

Base Classifier

- [InvisibleStereotype](#)

437. NAV-2

The Integrated Dictionary (NAV-2) provides definitions of all terms used throughout the architectural data.

To fill in NAV-2 table, UPDM elements have to be added to it.

*Click **Add Element** button to create a new or to add an existing element to the table.*

"Name", "Definition", "Alias", "Same As", "Documentation", and wide range of column cells are allowed to edit in the table. "UPDM Type", "UML Metatype", "SysML Type", "BPMN Type" cells are read only.

Rows (UPDM Elements) can be removed from the model or only from the table, can be ordered, and exported to the CSV or HTML. Four kind of reports can be printed reflecting the data shown in the table.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NAV-2	\$NAV-2 [i] .hideColumns

438. NAV-2 Package

Base Classifier

- [InvisibleStereotype](#)

439. NAV-2 Report

Base Classifier

- [InvisibleStereotype](#)

440. NAV-2 Tabular Report

Base Classifier

- [InvisibleStereotype](#)

441. NCV-1

Base Classifier

- [InvisibleStereotype](#)

442. NCV-1 Package**Base Classifier**

- [InvisibleStereotype](#)

443. NCV-2**Base Classifier**

- [InvisibleStereotype](#)

444. NCV-2 Package**Base Classifier**

- [InvisibleStereotype](#)

445. NCV-3**Base Classifier**

- [InvisibleStereotype](#)

446. NCV-3 Package**Base Classifier**

- [InvisibleStereotype](#)

447. NCV-4**Base Classifier**

- [InvisibleStereotype](#)

448. NCV-4 Package

Base Classifier

- [InvisibleStereotype](#)

449. NCV-5

NCV-5 Capability to Organisational Deployment Mapping shows deployment of capability configurations to specific organizations during a specific Enterprise Phase.

*Click **Add Rows** button to select Actual Organizational Resources. Selected Actual Organizational Resources shall be added as table rows.*

*Click **Add/Remove Columns** button to select or deselect Capabilities displayed as Columns in the table.*

Table cells are allowed to edit in the table. System Resources can be added to the table as cells contents. Added resources will be marked as exhibiting Capability represented by column and used by Actual Organizational Resource represented by Row in the particular Enterprise time frame.

Rows (Actual Organization Resources) can be removed from model or only from table, can be ordered, exported to the CSV or HTML and NCV-5 Spreadsheet Report can be generated.

More Actions are available by clicking right mouse button on a cell

Base Classifier

- [InvisibleStereotype](#)
- [StV-5](#)

450. NCV-5 Package

Base Classifier

- [InvisibleStereotype](#)

451. NCV-5 Report

Base Classifier

- [InvisibleStereotype](#)

452. NCV-6

The Operational Activity to Capability Mapping (NCV-6) describes the mapping between the capabilities required by an Enterprise and the operational activities that those capabilities support.

The Rows of this matrix are Capabilities and the Columns are Standard Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Capabilities);*
- 2. Specify Columns scope (Standard Operational Activities);*
- 3. Click "Refresh" button.*

Standard Operational Activities maps to Capabilities using "Maps to Capability" relationship.

To map Standard Operational Activity to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

453. NCV-6 Package

Base Classifier

- [InvisibleStereotype](#)

454. NCV-7

The Capability to Services Mapping Matrix (NCV-7) depicts which Service Interfaces contribute to the achievement of a Capability. The Rows of this matrix are Service Interfaces and the Columns are Capabilities.

To build the Matrix:

1. **Specify Rows** scope (Service Interfaces);
2. **Specify Columns** scope (Capabilities);
3. Click "**Refresh**" button.

Service Interfaces expose Capabilities using "Expose" relationship.

To map Service Interface to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

455. NCV-7 Package

Base Classifier

- [InvisibleStereotype](#)

456. Needline

MODAF: A relationship between Nodes representing a bundle of InformationExchanges.

DoDAF: A needline documents the requirement to exchange information between nodes. The needline does not indicate how the information transfer is implemented.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Needline[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Needline[i].conformsTo

endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Needline[i].endBoundaryType</code>
Needline.end		Needline	<code>\$Needline[i].Needline.end</code>
propertySet	PropertySet	UPDMElement	<code>\$Needline[i].propertySet</code>
realizedExchange	OperationalExchange	Needline	<code>\$Needline[i].realizedExchange</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Needline[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Needline[i].URI</code>

457. NestedConnectorEnd

The *NestedConnectorEnd* stereotype of UML *ConnectorEnd* extends a UML *ConnectorEnd* so that the connected property may be identified by a multi-level path of accessible properties from the block that owns the connector.

Base Classifier

- [ElementPropertyPath](#)
- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
propertyPath	Property	ElementPropertyPath	<code>\$NestedConnectorEnd[i].propertyPath</code>

458. NoBuffer

When this stereotype is applied to object nodes, tokens arriving at the node are discarded if they are refused by outgoing edges, or refused by actions for object nodes that are input pins. This is typically used with fast or continuously flowing data values, to prevent buffer overrun, or to model transient values, such as electrical signals. For object nodes that are the target of continuous flows, «nobuffer» and «overwrite» have the same effect. The stereotype does not override UML token offering semantics; it just indicates what happens to the token when it is accepted. When the stereotype is not applied, the semantics are as in UML, specifically, tokens arriving at an object node that are refused by outgoing edges, or action for input pins, are held until they can leave the object node.

459. Node

MODAF: A Node (*MODAF::NodeType*) is a logical entity that performs operational activities. Note: nodes are specified independently of any physical realization.

DoDAF: A Node (*DoDAF::OperationalNode*) is an element of the operational architecture that produces, consumes, or processes information. NOTE: This is also a specialization of *Performer*.

Base Classifier

- [ActivitySubject](#)
- [NodeParent](#)
- [SubjectOfOperationalConstraint](#)
- [SubjectOfOperationalStateMachine](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$Node[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Node[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Node[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Node[i].conformsTo</code>
connectedNodes	Node	Node	<code>\$Node[i].connectedNodes</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Node[i].endBoundaryType</code>
Node.ownedPort		Node	<code>\$Node[i].Node.ownedPort</code>
Node.performs		Node	<code>\$Node[i].Node.performs</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$Node[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$Node[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$Node[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Node[i].startBoundaryType</code>
SubjectOfOperationalStateMachine.ownedBehavior		SubjectOfOperationalStateMachine	<code>\$Node[i].SubjectOfOperationalStateMachine.ownedBehavior</code>
URI	String	UPDMElement	<code>\$Node[i].URI</code>

460. Node Impact Analysis Map

The *Node Impact Analysis Map* depicts the model elements influenced by the changes of the Node. The predefined map includes:

- *Performs Operational Activity*
- *Exhibits Capability, Context (Node is used as Context)*
- *Inputs and Outputs (Nodes connected using Operational Exchanges)*
- *Operational Exchanges*

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:

Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node /.

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

461. Node Role Impact Analysis Map

The **Node Role Impact Analysis Map** depicts the model elements influenced by the changes of the Node Role. The predefined map includes:

- Context (Nodes that are the context of the Node Role)
- Inputs and Outputs (Node Roles connected using Operational Exchanges)
- Performs In Context (Operational Activities performed by the Node Role)
- Performs (by Type) (Operational Activities performed by the Type of the Node Role)
- Exhibits Capability (by Type) (Capabilities exhibited by the Type of the Node Role)

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:


-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:

Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .


Expand / suppress branches - click on smart manipulator after the Node .

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- [InvisibleStereotype](#)

462. NodeAssociation

Relationship summarizing Operational Exchanges between connected Nodes. It is not a part of UPDM 2.0 specification.

463. NodeConceptRole

Base Classifier

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$NodeConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$NodeConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$NodeConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$NodeConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$NodeConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$NodeConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$NodeConceptRole[i].URI</code>

464. NodeOperation

UPDM: A partial or full realization of an OperationalActivity.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$NodeOperation[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$NodeOperation[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$NodeOperation[i].endBoundaryType</code>

NodeOperation.ownedParameter		NodeOperation	\$NodeOperation[i].NodeOperation.ownedParameter
propertySet	PropertySet	UPDMElement	\$NodeOperation[i].propertySet
realizes	OperationalActivity	NodeOperation	\$NodeOperation[i].realizes
startBoundaryType	ISO8601DateTime	UPDMElement	\$NodeOperation[i].startBoundaryType
URI	String	UPDMElement	\$NodeOperation[i].URI

465. NodeParent

UPDM: An abstract element representing the owners/context of composite structure at the operational level.

MODAF: The abstract supertype of all elements that can have child Nodes (LogicalArchitecture, ProblemDomain & NodeType)

DoDAF: NA

Base Classifier

- Participant

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	\$NodeParent[i].actsUpon
actualPropertySet	ActualPropertySet	UPDMElement	\$NodeParent[i].actualPropertySet
appliesTo	UPDMElement	PropertySet	\$NodeParent[i].appliesTo
conformsTo	Standard	UPDMElement	\$NodeParent[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$NodeParent[i].endBoundaryType
physicalLocation	ActualLocation	LocationHolder	\$NodeParent[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$NodeParent[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$NodeParent[i].requiredEnvironment
startBoundaryType	ISO8601DateTime	UPDMElement	\$NodeParent[i].startBoundaryType
URI	String	UPDMElement	\$NodeParent[i].URI

466. NodePort

UPDM: A port is a property of a Node that specifies a distinct interaction point between the node and its environment or between the (behavior of the) node and its internal parts. It is the “entry/exit” point where resources (e.g., energy, information/data and people, etc) flow in and out of a node.

Base Classifier

- UPDMElement

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$NodePort[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$NodePort[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$NodePort[i].endBoundaryType
NodePort.type		NodePort	\$NodePort[i].NodePort.type

propertySet	PropertySet	UPDMElement	<code>\$NodePort[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$NodePort[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$NodePort[i].URI</code>

467. NodeRole

MODAF: A NodeRole (MODAF::Node) is used to link a parent Node to its sub-nodes.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$NodeRole[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$NodeRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$NodeRole[i].endBoundaryType</code>
NodeRole.class		NodeRole	<code>\$NodeRole[i].NodeRole.class</code>
NodeRole.type		NodeRole	<code>\$NodeRole[i].NodeRole.type</code>
performsInContext	OperationalActivity	NodeRole	<code>\$NodeRole[i].performsInContext</code>
propertySet	PropertySet	UPDMElement	<code>\$NodeRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$NodeRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$NodeRole[i].URI</code>

468. NoLongerUsedMilestone

MODAF: Asserts that an ActualOrganisationResource ceased to use or is slated to cease using a CapabilityConfiguration from a specific point in time. --This is used to describe capabilities going out of service with specific organisations or posts.

DoDAF:NA

Base Classifier

- [ActualProjectMilestone](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualProjectMilestone.classifier		ActualProjectMilestone	<code>\$NoLongerUsedMilestone[i].ActualProjectMilestone.classifier</code>
ActualProjectMilestone.slot		ActualProjectMilestone	<code>\$NoLongerUsedMilestone[i].ActualProjectMilestone.slot</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$NoLongerUsedMilestone[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$NoLongerUsedMilestone[i].conformsTo</code>
date	ISO8601DateTime	ActualProjectMilestone	<code>\$NoLongerUsedMilestone[i].date</code>
description	String	ActualProjectMilestone	<code>\$NoLongerUsedMilestone[i].description</code>

endBoundaryType	ISO8601DateTime	UPDMElement	\$NoLongerUsedMilestone[i].endBoundaryType
noLongerUsedBy	ActualOrganizationalResource	NoLongerUsedMilestone	\$NoLongerUsedMilestone[i].noLongerUsedBy
propertySet	PropertySet	UPDMElement	\$NoLongerUsedMilestone[i].propertySet
resource	SystemResource	ActualProjectMilestone	\$NoLongerUsedMilestone[i].resource
startBoundaryType	ISO8601DateTime	UPDMElement	\$NoLongerUsedMilestone[i].startBoundaryType
URI	String	UPDMElement	\$NoLongerUsedMilestone[i].URI

469. nonStreaming

Used for activities that accept inputs only when they start, and provide outputs only when they finish.

470. Normal

Normal distribution - constant probability between min and max

Base Classifier

- DistributedProperty

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
mean	Real	Normal	\$Normal[i].mean
standardDeviation	Real	Normal	\$Normal[i].standardDeviation

471. Note

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Text	String	Note	\$Note[i].Text

472. NOV-1

Base Classifier

- InvisibleStereotype

473. NOV-1 Package

Base Classifier

- InvisibleStereotype

474. NOV-1i

Base Classifier

- InvisibleStereotype
- OV-1i

475. NOV-2

Base Classifier

- InvisibleStereotype

476. NOV-2 Package

Base Classifier

- InvisibleStereotype

477. NOV-2i

Base Classifier

- InvisibleStereotype
- OV-2i

478. NOV-3

Operational Information Requirements (NOV-3) addresses operational exchanges between nodes.

To fill in NOV-3 table, Existing Operational Exchanges of any available kind such as Information Exchange, Materiel Exchange, Organizational Exchange, Energy Exchange and etc. have to be added to it.

*Click **Add New** button to create a new Operational Exchange between selected Nodes.*

Click **Add Existing** button to select Operational Exchanges or Needlines. In case Needline is selected, all Operational Exchanges flowing via it will be added to the table.

"Operational Exchange Identifier", "Operational Exchange Item Name, Producing and Consuming Operational Activities, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Operational Exchanges NOV-2 product is recommended to use.

Rows (Operational Exchanges) can be removed from model or only from table, can be filtered according to represented Operational Exchange kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NOV-3	\$NOV-3 [i] .hideColumns

479. NOV-3 Package

Base Classifier

- [InvisibleStereotype](#)

480. NOV-3 Report

Base Classifier

- [InvisibleStereotype](#)

481. NOV-3 Role Based

The Role-based Operational Information Requirements (NOV-3) addresses the resources exchanged between node roles and the relevant attributes of the exchanges To fill in NOV-3 table, Existing Operational Exchanges of any available kind such as Information Exchange, Materiel Exchange, Organizational Exchange, Energy Exchange, Geo Political Extent Exchange and etc. have to be added to it.

Click **Add Existing** button to select *Operational Exchanges*, *Needlines*, or *Service Channels*. In case *Needline* or *Service Channel* is selected, all *Operational Exchanges* flowing via it will be added to the table.

"Operational Exchange ID", "Operational Exchange Item", "Producing and Consuming Operational Activities", and wide range of measurement cells are allowed to edit in the table. For creation and modification of *Operational Exchanges NOV-2* internal diagram is recommended to use.

Rows (*Operational Exchanges*) can be removed from model or only from table, can be filtered according to represented *Operational Exchange* kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NOV-3 Role Based	\$NOV-3RoleBased[i].hideColumns

482. NOV-3 Role Based Report

Base Classifier

- [InvisibleStereotype](#)

483. NOV-4

Base Classifier

- [InvisibleStereotype](#)

484. NOV-4 Package

Base Classifier

- [InvisibleStereotype](#)

485. NOV-5

Base Classifier

- [InvisibleStereotype](#)

486. NOV-5 Package

Base Classifier

- [InvisibleStereotype](#)

487. NOV-6a

The Operational Rule Model (NOV-6a) specifies operational or business rules that are constraints on an enterprise, a mission, operation, business, or architecture.

There are two ways to fill this table.

- 1. Add new Operational Constraint. Click **Add New** button and select constrained Node, Operational Activity, Entity Item, Mission, Exchange Element, or Operational Exchange). Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. Add Existing Operational Constraints. Click **Add Existing** button and select Operational Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Operational Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NOV-6a	\$NOV-6a[i].hideColumns

488. NOV-6a Package

Base Classifier

- [InvisibleStereotype](#)

489. NOV-6a Report

Base Classifier

- [InvisibleStereotype](#)

490. NOV-6b

Base Classifier

- [InvisibleStereotype](#)

491. NOV-6b Package

Base Classifier

- [InvisibleStereotype](#)

492. NOV-6c

Base Classifier

- [InvisibleStereotype](#)

493. NOV-6c Package

Base Classifier

- [InvisibleStereotype](#)

494. NOV-7

Base Classifier

- [InvisibleStereotype](#)

495. NOV-7 Package

Base Classifier

- [InvisibleStereotype](#)

496. NPV-1

The Responsibility Matrix (NPV-1) describes the mapping between the Actual Projects and the Actual Organizational Resources. The Rows of this matrix are Actual Projects and the Columns are Actual Organizational Resources (Actual Organization or Actual Post).

To build the Matrix:

- 1. Specify Rows scope (Actual Projects);*
- 2. Specify Columns scope (Actual Organizations and Actual Posts);*
- 3. Click "Refresh" button.*

Actual Organizational Resources maps to Actual Projects by "responsibleFor" property.

To map Actual Project to Actual Organizational Resource, Click on the intersection between the desired elements. By pressing on the intersection one more time, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

497. NPV-1 Package

Base Classifier

- [InvisibleStereotype](#)

498. NPV-1c

Base Classifier

- [InvisibleStereotype](#)

499. NPV-2

Programme to Capability Mapping (NPV-2) describes the mapping of programs and projects to capabilities to show how the specific projects and program elements help to achieve a capability.

The Rows of this matrix are Capabilities and the Columns are Projects.

To build the Matrix:

- 1. Specify Rows scope (Capabilities);*
- 2. Specify Columns scope (Projects);*
- 3. Click "Refresh" button.*

There are two possible ways to map Capability and a Project:

- 1. Activity is part of Project ("Activity Part of Project" relationship and a part of Capability ("Activity Part of Capability" relationship).*
- 2. Project owns Increment Milestones (Owned Milestones property) that are related to Resources (Resources property) exhibiting (Capability of Performer relationship) capabilities.*

Both transitive structures of relations map Capabilities to Projects. The mapping is displayed in the Matrix.

Base Classifier

- [InvisibleStereotype](#)

500. NPV-2 Package

Base Classifier

- [InvisibleStereotype](#)

501. NSOV-1

Base Classifier

- [InvisibleStereotype](#)

502. NSOV-1 Package

Base Classifier

- [InvisibleStereotype](#)

503. NSOV-2

Base Classifier

- [InvisibleStereotype](#)

504. NSOV-2 Package

Base Classifier

- [InvisibleStereotype](#)

505. NSOV-2 Report

Base Classifier

- [InvisibleStereotype](#)

506. NSOV-3

Services to Operational Activities Mapping (NSOV-3) purpose is to provide traceability by illustrating which services support which operational activities..

The Rows of this matrix are Service Interfaces and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Service Interfaces);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Operational Activity maps to Capability ("Maps To Capability" relationship) that is exposed by Service Interface ("Expose" relationship).

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NSOV-3	\$NSOV-3[i].hideColumns

507. NSOV-3 Package

Base Classifier

- [InvisibleStereotype](#)

508. NSOV-4

Base Classifier

- [InvisibleStereotype](#)

509. NSOV-4 Package

Base Classifier

- [InvisibleStereotype](#)

510. NSOV-5

Base Classifier

- [InvisibleStereotype](#)

511. NSOV-5 Package

Base Classifier

- [InvisibleStereotype](#)

512. NSV-1

Base Classifier

- [InvisibleStereotype](#)

513. NSV-1 Package

Base Classifier

- [InvisibleStereotype](#)

514. NSV-10a

Systems Rule Model (NSV-10a) allows you to constraint Systems View Architectural elements.

There are two ways to fill this table:

- 1. Add new Resource Constraint. Click **Add New** button and select Systems Element (Resource Artifact, Software, Capability Configuration, Organization, Post, Function, Exchange Element, Entity Item, Resource Interaction) to be constrained. Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. Add Existing Resource Constraints. Click **Add Existing** button and select Resource Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Resource Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NSV-10a	<code>\$(NSV-10a[i]).hideColumns</code>

515. NSV-10a Package

Base Classifier

- [InvisibleStereotype](#)

516. NSV-10a Report**Base Classifier**

- [InvisibleStereotype](#)

517. NSV-10b**Base Classifier**

- [InvisibleStereotype](#)

518. NSV-10b Package**Base Classifier**

- [InvisibleStereotype](#)

519. NSV-10c**Base Classifier**

- [InvisibleStereotype](#)

520. NSV-10c Package**Base Classifier**

- [InvisibleStereotype](#)

521. NSV-11**Base Classifier**

- [InvisibleStereotype](#)

522. NSV-11 Package

Base Classifier

- [InvisibleStereotype](#)

523. NSV-12

Service Provision (NSV-12) Matrix defines the relationships between the Resources and Service Interfaces.

The Rows of this matrix are Service Interfaces and the Columns are System Resources.

To build the Matrix:

- 1. Specify Rows scope (Service Interfaces);*
- 2. Specify Columns scope (System Resources);*
- 3. Click "Refresh" button.*

Cells here represents Services or Requests. If the resource provides service, Service Icon is displayed. If it requests (consumes) service, Request Icon is displayed.

By double clicking on the empty cell new Service will be created. By double clicking once more new Request will be created and the Service will be deleted. By double clicking third time cell will be cleared and the Request will be deleted.

Base Classifier

- [InvisibleStereotype](#)

524. NSV-12 Package

Base Classifier

- [InvisibleStereotype](#)

525. NSV-1i

Base Classifier

- InvisibleStereotype
- SV-1i

526. NSV-2

Base Classifier

- InvisibleStereotype

527. NSV-2 Package

Base Classifier

- InvisibleStereotype

528. NSV-2i

Base Classifier

- InvisibleStereotype
- SV-2i

529. NSV-3

Systems to Systems Matrix (NSV-3) shows how resources interact to each other.

It is editable matrix where cells of the matrix represent Resource interactions and the headers represent System Resources.

Arrow can be on one side and on the other side. It depends on the direction of Resource interaction flow between System Resources. NSV-3 matrix consists of as much rows and columns as there are System Resources in the matrix data source.

To build the Matrix:

1. *Specify Rows scope (System Resources);*
2. *Specify Columns scope (System Resources);*
3. *Click "Refresh" button.*

By clicking on the cell context menu opens allowing to open new Resource Interaction creation wizard or delete existing relationships.

Base Classifier

- [InvisibleStereotype](#)

530. NSV-3 Package

Base Classifier

- [InvisibleStereotype](#)

531. NSV-4

Base Classifier

- [InvisibleStereotype](#)

532. NSV-4 Package

Base Classifier

- [InvisibleStereotype](#)

533. NSV-5

The Systems Function to Operational Activity Traceability Matrix (NSV-5) addresses the linkage between Functions described in NSV-4 and Operational Activities specified in NOV-5.

The Rows of this matrix are Functions and the Columns are Operational Activities.

To build the Matrix:

1. **Specify Rows** scope (Functions);
2. **Specify Columns** scope (Operational Activities);
3. Click "**Refresh**" button.

Functions implements Operational Activities using "Implements" relationship.

To map Function to Operational Activity, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

534. NSV-5 Package

Base Classifier

- [InvisibleStereotype](#)

535. NSV-6

The Systems Data Exchange Matrix (NSV-6) specifies the characteristics of the data exchanged between Resources.

To fill in NSV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add New** button to create a new Resource Interaction between selected System Resources.*

*Click **Add Existing** button and select Resource Interactions or Resource Interfaces. In case Resource Interface is selected, all Resource Interactions flowing via it will be added to the table.*

"Resource Interaction identifier". "Resource Interaction Item Name, Producing and Consuming Functions, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Resource Interactions NSV-1 product is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NSV-6	\$NSV-6[i].hideColumns

536. NSV-6 Package

Base Classifier

- [InvisibleStereotype](#)

537. NSV-6 Report

Base Classifier

- [InvisibleStereotype](#)

538. NSV-6 Role Based

The Role-based Systems Data Exchange Matrix (NSV-6) addresses specifies the characteristics of the data exchanged between resources.

To fill in NSV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add Existing** button to select Resource Interactions, Resource Interfaces, Resource Connectors, or Service Channels. In case Resource Interface, Resource Connector, or Service Channel is selected, all Resource Interactions flowing via it will be added to the table.*

"Interaction ID", "Resource Interaction Item", "Producing and Consuming Functions", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Resource Interactions SV-1 internal diagram is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be filtered according to the connector kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NSV-6 Role Based	\$NSV-6RoleBased[i].hideColumns

539. NSV-6 Role Based Report

Base Classifier

- [InvisibleStereotype](#)

540. NSV-7 Actual

System Actual Quality Requirements Description (NSV-7 Actual) depicts the Actual values of performance characteristics of a Resource.

There are three ways to add a row in this table:

- 1. Add new measurable Resource; Click **Add New** button and select one or more System Resources that have at least one Measurement Set Defined (see NSV-7 Typical). Specify values for each Measurement - directly in the table cells.*
- 2. Add existing Measures or measurable Resources. Click **Add Existing** button and select Existing Measurements or Resources.*
- 3. Add missing Measurements. Click “**Add the missing Measurements**” button to update table to include latest model changes.*

Rows (Actual Measurements) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NSV-7 Actual	\$NSV-7Actual[i].hideColumns

541. NSV-7 Package

Base Classifier

- [InvisibleStereotype](#)

542. NSV-7 Report

Base Classifier

- [InvisibleStereotype](#)

543. NSV-7 Typical

System Typical Quality Requirements Description (NSV-7 Typical) is the and depicts the possible types of performance characteristics of a Resource.

There are two ways to add a row in this table:

- 1. Add new Measurement Set. Click **Add New** button and select the owning element for Measurement Set. Specify Measurements to the Measurement Set and Resources to be Measured - straight in the table cells.*
- 2. Add existing Measurements Sets. Click **Add Existing** button and select Existing Measurement Sets.*

Rows (Measurement Sets) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	NSV-7 Typical	\$NSV-7Typical[i].hideColumns

544. NSV-8

Base Classifier

- [InvisibleStereotype](#)
- [SV-8](#)

545. NSV-8 Package

Base Classifier

- [InvisibleStereotype](#)

546. NSV-8 Report**Base Classifier**

- [InvisibleStereotype](#)

547. NSV-9

The Technology Forecast (NSV-9) defines the underlying current and expected supporting technologies and skills.

Three major steps should be done to create the table:

1. Add Rows to the Table. There are two ways to add a row to this table:

*1.1. Add new Resource as row Header. Click **Add New** button and select System Resource you want to create. Specify owner for selected Resource.*

*1.2. Add Existing Resource as row Header. Click **Add Existing** button and select one or more existing System Resources.*

*2. Add columns to the table. Click "**Add/Remove forecast**" button. Specify time periods for the forecasting: select or create Time Line Package to store forecast dates; select forecast kind. According to the selected forecast kind, specify additional options needed (see Forecasting Period Dialog help for more information).*

3. Fill in the cells with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select System Resources, Standards, Protocols or Competences to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected time period.

Rows (System Resources) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

548. NSV-9 Package

Base Classifier

- [InvisibleStereotype](#)

549. NTV-1

Technical Standard Profile (NTV-1) table defines the technical and non technical standards, guidance and policy applicable to the architecture.

There are two ways to add a row in this table:

1. Add new UPDM Element. Click “Add new UPDM Element” button. Select UPDM Element (this term means any of the UPDM available element). Specify the owner for the selected element.

2. Add Existing UPDM elements. Click “Add Existing UPDM Element” button and select UPDM elements.

For each row there should be assigned one or more Standards or Protocols that the particular UPDM element must conform to. Click “...” button on any cell in the “Standard/Policy” column.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

550. NTV-1 Package**Base Classifier**

- [InvisibleStereotype](#)

551. NTV-2

Technical Standards Forecast (NTV-2) table defines expected changes in technology related standards and conventions.

You will find this table identical to the NSV-9 table. In general they are identical in implementation, but NTV-2 is more likely to be used for Standards and Protocols forecasting.

Three major steps should be done to create the table:

1. Add Rows to the Table. There are two ways to add a row to the table:

1.1. Add new Subject of Forecast as row Header. Click Add New button and select element you want to create. Specify owner for selected element.

1.2. Add Existing Subject of Forecast as row Header. Click **Add Existing** button and select one or more existing elements.

2. Add columns to the table. Click "**Add/Remove forecast**" button. Specify Time Periods for the forecasting: select or create Time Line Package to store Forecast dates; select forecast kind. According to the selected Forecast Kind, specify additional options needed (see Forecasting Period Dialog for more info).

3. Fill in the cells with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select Resources, Standards, Protocols or Competences to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected Time Period.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

552. NTV-2 Package

Base Classifier

- [InvisibleStereotype](#)

553. Number

554. NumberingScheme

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
numberingStyle	NumberingStyle	NumberingScheme	\$NumberingScheme[i].numberingStyle

555. NumberingStyle

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
nested	Enumeration Literal	NumberingStyle	\$NumberingStyle[i].nested
normal	Enumeration Literal	NumberingStyle	\$NumberingStyle[i].normal

556. NumberingStyle

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Consecutive	Enumeration Literal	NumberingStyle	\$NumberingStyle[i].Consecutive
Multi-level	Enumeration Literal	NumberingStyle	\$NumberingStyle[i].Multi-level

557. NumberOwner

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
customNumberingData	String	NumberOwner	\$NumberOwner[i].customNumberingData

558. numberOwner

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
numberingStyle	NumberingStyle	numberOwner	\$numberOwner[i].numberingStyle
prefix	ReqNumber	numberOwner	\$numberOwner[i].prefix
separator	ReqNumber	numberOwner	\$numberOwner[i].separator

559. NumberPart

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
expression	String	NumberPart	\$NumberPart[i].expression

initialValue	String	NumberPart	\$NumberPart[i].initialValue
sequence	SequenceType	NumberPart	\$NumberPart[i].sequence

560. objectiveFunction

An objective function (aka optimization or cost function) is used to determine the overall value of an alternative in terms of weighted criteria and/or moe's.

561. ObjectNodeOrderingKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
FIFO	Enumeration Literal	ObjectNodeOrderingKind	\$ObjectNodeOrderingKind[i].FIFO
LIFO	Enumeration Literal	ObjectNodeOrderingKind	\$ObjectNodeOrderingKind[i].LIFO
ordered	Enumeration Literal	ObjectNodeOrderingKind	\$ObjectNodeOrderingKind[i].ordered
unordered	Enumeration Literal	ObjectNodeOrderingKind	\$ObjectNodeOrderingKind[i].unordered

562. OclState

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
<>	\$OCL_Boolean	OclState	\$OclState[i].<>
=	\$OCL_Boolean	OclState	\$OclState[i].=

563. OntologyReference

MODAF: A reference to an element in a recognized external ontology or taxonomy.

DoDAF:NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$OntologyReference[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$OntologyReference[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$OntologyReference[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$OntologyReference[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$OntologyReference[i].startBoundaryType
URI	String	UPDMElement	\$OntologyReference[i].URI

url	String	OntologyReference	!OntologyReference[i].url
-----	--------	-------------------	---------------------------

564. Operational Activity Implementation Map

The **Operational Activity Implementation Map** depicts the behavioral implementation of an Operational Activity, relating the **Operational Activity** and **Functions** that implement the operational behavior. When the Operational Activity has composite behaviors and when Functions have composite functionality, the Operational Activity Implementation Map also includes the behavioral elements for the composed elements.

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:


Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node /.

Move the whole structure - click on the empty place in the Relation Map and drag.


Move the selected Node - click on the Node and drag.

Insert New Element - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.

Insert New Element of the Same Type - type the name and press Ctrl+Enter to create an element of the same type.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

565. Operational Activity Map

The **Operational Activity Map** depicts the behavioral decomposition of an **Operational Activity**.

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:


-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:


Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node .

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Insert New Element - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.

Insert New Element of the Same Type - type the name and press Ctrl+Enter to create an element of the same type.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

566. Operational View MODAF

Base Classifier

- InvisibleStereotype

567. Operational Viewpoint

Base Classifier

- [InvisibleStereotype](#)

568. OperationalAction**Base Classifier**

- [InvisibleStereotype](#)

569. OperationalActivity

MODAF: A logical process, specified independently of how the process is carried out. DoDAF: An activity is an action performed in conducting the business of an enterprise. It is a general term that does not imply a placement in a hierarchy (e.g., it could be a process or a task as defined in other documents and it could be at any level of the hierarchy of the OV-5). It is used to portray operational actions not hardware/software system functions. NOTE: This is also a specialization of Activity.

DoDAF:NA

Base Classifier

- [Activity](#)
- [Process](#)
- [SubjectOfOperationalConstraint](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
activityPerformableUnderCondition	Environment	Activity	<code>\$OperationalActivity[i].activityPerformableUnderCondition</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OperationalActivity[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$OperationalActivity[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalActivity[i].endBoundaryType</code>
OperationalActivity.ownedParameter		OperationalActivity	<code>\$OperationalActivity[i].OperationalActivity.ownedParameter</code>
propertySet	PropertySet	UPDMElement	<code>\$OperationalActivity[i].propertySet</code>
realizedBy	NodeOperation	OperationalActivity	<code>\$OperationalActivity[i].realizedBy</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalActivity[i].startBoundaryType</code>
subject	ActivitySubject	OperationalActivity	<code>\$OperationalActivity[i].subject</code>
URI	String	UPDMElement	<code>\$OperationalActivity[i].URI</code>

570. OperationalActivityAction

UPDM: The OperationalActivityAction is defined as a call behavior action that invokes the activity that needs to be preformed.

MODAF: Used to relate an OperationalActivity to its sub-activities.

DoDAF:NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OperationalActivityAction[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$OperationalActivityAction[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalActivityAction[i].endBoundaryType</code>
OperationalActivityAction.activity		OperationalActivityAction	<code>\$OperationalActivityAction[i].OperationalActivityAction.activity</code>
OperationalActivityAction.behavior		OperationalActivityAction	<code>\$OperationalActivityAction[i].OperationalActivityAction.behavior</code>
propertySet	PropertySet	UPDMElement	<code>\$OperationalActivityAction[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalActivityAction[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$OperationalActivityAction[i].URI</code>

571. OperationalActivityEdge

UPDM An extension of <<ActivityEdge>> that is used to model the flow of control/objects through an OperationalActivity.

MODAF: An OperationalActivityEdge (MODAF::OperationalActivityFlow) is a flow of information, energy or materiel from one activity to another.

DoDAF:NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OperationalActivityEdge[i].actualPropertySet</code>
carriedItem	OperationalExchangeItem	OperationalActivityEdge	<code>\$OperationalActivityEdge[i].carriedItem</code>
conformsTo	Standard	UPDMElement	<code>\$OperationalActivityEdge[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalActivityEdge[i].endBoundaryType</code>
OperationalActivityEdge.owner		OperationalActivityEdge	<code>\$OperationalActivityEdge[i].OperationalActiv</code>

			ityEdge.owner
propertySet	PropertySet	UPDMElement	\$OperationalActivityEdge[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalActivityEdge[i].startBoundaryType
URI	String	UPDMElement	\$OperationalActivityEdge[i].URI

572. OperationalConstraint

UPDM: An abstract Class that is extended by OperationalConstraint (A rule governing an operational behaviour or property.) and ResourceConstraint.

MODAF: A rule governing an operational behaviour or property.

DoDAF: A principle or condition that governs behavior; a prescribed guide for conduct or action (Rule).

Base Classifier

- [Rule](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$OperationalConstraint[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$OperationalConstraint[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalConstraint[i].endBoundaryType
OperationalConstraint.constrainedElement		OperationalConstraint	\$OperationalConstraint[i].OperationalConstraint.constrainedElement
propertySet	PropertySet	UPDMElement	\$OperationalConstraint[i].propertySet
ruleKind	RuleKind	Rule	\$OperationalConstraint[i].ruleKind
startBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalConstraint[i].startBoundaryType
URI	String	UPDMElement	\$OperationalConstraint[i].URI

573. OperationalEventTrace

MODAF: An OperationalEventTrace (MODAF::OperationalInteractionSpecification) is a specification of the interactions between nodes in an operational architecture.

DoDAF: The Operational Event-Trace Description (OV-6c) DoDAF-described View provides a time ordered examination of the resource flows as a result of a particular scenario. Each event-trace diagram will have an accompanying description that defines the particular scenario or situation.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$OperationalEventTrace[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$OperationalEventTrace[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalEventTrace[i].endBoundaryType
OperationalEventTrace.message		OperationalEventTrace	\$OperationalEventTrace[i].OperationalEventTrace.message

OperationalEventTrace.owner		OperationalEventTrace	\$OperationalEventTrace[i].OperationalEventTrace.owner
propertySet	PropertySet	UPDMElement	\$OperationalEventTrace[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalEventTrace[i].startBoundaryType
URI	String	UPDMElement	\$OperationalEventTrace[i].URI

574. OperationalExchange

UPDM: An utility element used as common flow for:

- *InformationExchange*
- *OrganizationalExchange*
- *EnergyExchange*
- *MaterielExchange*
- *ConfigurationExchange*
- *GeoPoliticalExtent*

An operational exchange is formed when an activity of one operational node consumes items produced by another activity of a different operational node.

An operational exchange describes the characteristics of the exchanged item, such as the content, format (voice, imagery, text and message format, etc.), throughput requirements, security or classification level, timeliness requirement, and the degree of interoperability.

MODAF: An OperationalExchange (MODAF::LogicalFlow) asserts that a flow exists or is required between Nodes (e.g. flows of information, people, materiel, or energy).

Base Classifier

- [Exchange](#)
- [SubjectOfOperationalConstraint](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$OperationalExchange[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$OperationalExchange[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalExchange[i].endBoundaryType
exchangeKind	OperationalExchangeKind	OperationalExchange	\$OperationalExchange[i].exchangeKind
OperationalExchange.conveyed		OperationalExchange	\$OperationalExchange[i].OperationalExchange.conveyed
OperationalExchange.informationSource		OperationalExchange	\$OperationalExchange[i].OperationalExchange.informationSource
OperationalExchange.informationTarget		OperationalExchange	\$OperationalExchange[i].OperationalExchange.informationTarget
OperationalExchange.realization/realizingConnector		OperationalExchange	\$OperationalExchange[i].OperationalExchange.realization/realizingConnector
OperationalExchange.realizingAct		OperationalExchange	\$OperationalExchange[i].OperationalExchange.

ivityEdge			realizingActivityEdge
OperationalExchange.realizingMessage		OperationalExchange	\$OperationalExchange[i].OperationalExchange.realizingMessage
propertySet	PropertySet	UPDMElement	\$OperationalExchange[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalExchange[i].startBoundaryType
URI	String	UPDMElement	\$OperationalExchange[i].URI

575. OperationalExchangeItem

UPDM An abstract utility element used as common ancestor for:

- *InformationElement*
- *ResourceArtifact*
- *Energy*
- *OrganizationalResource*
- *CapabilityConfiguration*
- *GeoPoliticalExtent*

Base Classifier

- [ActivitySubject](#)
- [Resource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	\$OperationalExchangeItem[i].actsUpon
actualPropertySet	ActualPropertySet	UPDMElement	\$OperationalExchangeItem[i].actualPropertySet
appliesTo	UPDMElement	PropertySet	\$OperationalExchangeItem[i].appliesTo
conformsTo	Standard	UPDMElement	\$OperationalExchangeItem[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalExchangeItem[i].endBoundaryType
physicalLocation	ActualLocation	LocationHolder	\$OperationalExchangeItem[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$OperationalExchangeItem[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$OperationalExchangeItem[i].requiredEnvironment
startBoundaryType	ISO8601DateTime	UPDMElement	\$OperationalExchangeItem[i].startBoundaryType
URI	String	UPDMElement	\$OperationalExchangeItem[i].URI

576. OperationalExchangeKind

Enumeration of operational exchange kinds, used to support the exchangeKind tag of the OperationalExchange stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ConfigurationExchange	Enumeration Literal	OperationalExchangeKind	<code>\$OperationalExchangeKind[i].ConfigurationExchange</code>
EnergyExchange	Enumeration Literal	OperationalExchangeKind	<code>\$OperationalExchangeKind[i].EnergyExchange</code>
GeoPoliticalExtentExchange	Enumeration Literal	OperationalExchangeKind	<code>\$OperationalExchangeKind[i].GeoPoliticalExtentExchange</code>
InformationExchange	Enumeration Literal	OperationalExchangeKind	<code>\$OperationalExchangeKind[i].InformationExchange</code>
MaterielExchange	Enumeration Literal	OperationalExchangeKind	<code>\$OperationalExchangeKind[i].MaterielExchange</code>
OrganizationalExchange	Enumeration Literal	OperationalExchangeKind	<code>\$OperationalExchangeKind[i].OrganizationalExchange</code>

577. OperationalMessage

UPDM: Message for use in an Operational Event-Trace which carries any of the subtypes of OperationalExchange. This is used to provide additional information about OperationalMessages for display on an OV-6c.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OperationalMessage[i].actualPropertySet</code>
carries	OperationalExchange	OperationalMessage	<code>\$OperationalMessage[i].carries</code>
conformsTo	Standard	UPDMElement	<code>\$OperationalMessage[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalMessage[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$OperationalMessage[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalMessage[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$OperationalMessage[i].URI</code>

578. OperationalParameter

UPDM Represents inputs and outputs of an OperationalActivity. It is typed by OperationalExchangeItem.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OperationalParameter[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$OperationalParameter[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalParameter[i].endBoundaryType</code>
OperationalParameter.type		OperationalParameter	<code>\$OperationalParameter[i].OperationalParameter</code>

			r.type
propertySet	PropertySet	UPDMElement	<code>\$OperationalParameter[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalParameter[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$OperationalParameter[i].URI</code>

579. OperationalState

Base Classifier

- [DesiredState](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OperationalState[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$OperationalState[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalState[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$OperationalState[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalState[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$OperationalState[i].URI</code>

580. OperationalStateDescription

UPDM: A state machine describing an operational behavior or property.

MODAF: An OperationalStateMachine (MODAF::OperationalStateDescription) is a rule governing an operational behaviour or property.

DoDAF: The Operational State Transition Description (OV-6b) DoDAF-described View is a graphical method of describing how an Operational Activity responds to various events by changing its state. The diagram represents the sets of events to which the Architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OperationalStateDescription[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$OperationalStateDescription[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalStateDescription[i].endBoundaryType</code>
OperationalStateDescription.owner		OperationalStateDescription	<code>\$OperationalStateDescription[i].OperationalStateDescription.owner</code>
propertySet	PropertySet	UPDMElement	<code>\$OperationalStateDescription[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OperationalStateDescription[i].startBoundaryType</code>

			yType
URI	String	UPDMElement	\$OperationalStateDescription[i].URI

581. Optional

When the «optional» stereotype is applied to parameters, the lower multiplicity must be equal to zero. This means the parameter is not required to have a value for the activity or any behavior to begin or end execution. Otherwise, the lower multiplicity must be greater than zero, which is called “required.”

582. Organization

MODAF: A group of persons, associated for a particular purpose.

DoDAF: A type of Organization.

Base Classifier

- [OrganizationalResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	\$Organization[i].actsUpon
actualPropertySet	ActualPropertySet	UPDMElement	\$Organization[i].actualPropertySet
affectedFunctions	Function	ResourceInteractionItem	\$Organization[i].affectedFunctions
appliesTo	UPDMElement	PropertySet	\$Organization[i].appliesTo
conformsTo	Standard	UPDMElement	\$Organization[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Organization[i].endBoundaryType
milestone	ActualProjectMilestone	SystemResource	\$Organization[i].milestone
physicalLocation	ActualLocation	LocationHolder	\$Organization[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$Organization[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$Organization[i].requiredEnvironment
Resource.ownedOperation		SystemResource	\$Organization[i].Resource.ownedOperation
Resource.ownedPort		SystemResource	\$Organization[i].Resource.ownedPort
Resource.performs		SystemResource	\$Organization[i].Resource.performs
startBoundaryType	ISO8601DateTime	UPDMElement	\$Organization[i].startBoundaryType
URI	String	UPDMElement	\$Organization[i].URI

583. Organization Structure Map

The **Organization Structure Map** depicts the structural composition of an Actual Organization. The predefined map includes: the **Actual Organization** itself, **Individual Person Roles** (or MODAF Actual Posts), and **Actual Person** elements. The Organization Structure Map also illustrates the relationships between these organizational elements.

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:


-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:

Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .


Expand / suppress branches - click on smart manipulator after the Node .

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- [InvisibleStereotype](#)

584. OrganizationalProjectRelationship

MODAF: A relationship between an ActualOrganisation and a Project.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OrganizationalProjectRelationship[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$OrganizationalProjectRelationship[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OrganizationalProjectRelationship[i].endBoundaryType</code>
endDate	ISO8601DateTime	OrganizationalProjectRelationship	<code>\$OrganizationalProjectRelationship[i].endDate</code>

			e
OrganizationalProjectRelationship.client		OrganizationalProjectRelationship	\$OrganizationalProjectRelationship[i].OrganizationalProjectRelationship.client
OrganizationalProjectRelationship.supplier		OrganizationalProjectRelationship	\$OrganizationalProjectRelationship[i].OrganizationalProjectRelationship.supplier
propertySet	PropertySet	UPDMElement	\$OrganizationalProjectRelationship[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$OrganizationalProjectRelationship[i].startBoundaryType
startDate	ISO8601DateTime	OrganizationalProjectRelationship	\$OrganizationalProjectRelationship[i].startDate
URI	String	UPDMElement	\$OrganizationalProjectRelationship[i].URI

585. OrganizationalResource

UPDM An abstract element that represents Organizations and Posts.

MODAF: Either an organization, or a post.

Base Classifier

- PhysicalResource

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	\$OrganizationalResource[i].actsUpon
actualPropertySet	ActualPropertySet	UPDMElement	\$OrganizationalResource[i].actualPropertySet
affectedFunctions	Function	ResourceInteractionItem	\$OrganizationalResource[i].affectedFunctions
appliesTo	UPDMElement	PropertySet	\$OrganizationalResource[i].appliesTo
conformsTo	Standard	UPDMElement	\$OrganizationalResource[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$OrganizationalResource[i].endBoundaryType
milestone	ActualProjectMilestone	SystemResource	\$OrganizationalResource[i].milestone
physicalLocation	ActualLocation	LocationHolder	\$OrganizationalResource[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$OrganizationalResource[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$OrganizationalResource[i].requiredEnvironment
Resource.ownedOperation		SystemResource	\$OrganizationalResource[i].Resource.ownedOperation
Resource.ownedPort		SystemResource	\$OrganizationalResource[i].Resource.ownedPort
Resource.performs		SystemResource	\$OrganizationalResource[i].Resource.performs
startBoundaryType	ISO8601DateTime	UPDMElement	\$OrganizationalResource[i].startBoundaryType
URI	String	UPDMElement	\$OrganizationalResource[i].URI

586. OrganizationConceptRole

Base Classifier

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OrganizationConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$OrganizationConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$OrganizationConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OrganizationConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$OrganizationConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OrganizationConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$OrganizationConceptRole[i].URI</code>

587. OrganizationType

DoDAF: A type of Organization.

Base Classifier

- [Organization](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$OrganizationType[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OrganizationType[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$OrganizationType[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$OrganizationType[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$OrganizationType[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OrganizationType[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$OrganizationType[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$OrganizationType[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$OrganizationType[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$OrganizationType[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$OrganizationType[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$OrganizationType[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$OrganizationType[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OrganizationType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$OrganizationType[i].URI</code>

588. OutOfServiceMilestone

MODAF: An OutOfServiceMilestone (MODAF::OutOfService) is a ProjectMilestone that indicates a project's deliverable is to go out of service.

DoDAF: NA

Base Classifier

- [ActualProjectMilestone](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualProjectMilestone.classifier		ActualProjectMilestone	\$OutOfServiceMilestone[i].ActualProjectMilestone.classifier
ActualProjectMilestone.slot		ActualProjectMilestone	\$OutOfServiceMilestone[i].ActualProjectMilestone.slot
actualPropertySet	ActualPropertySet	UPDMElement	\$OutOfServiceMilestone[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$OutOfServiceMilestone[i].conformsTo
date	ISO8601DateTime	ActualProjectMilestone	\$OutOfServiceMilestone[i].date
description	String	ActualProjectMilestone	\$OutOfServiceMilestone[i].description
endBoundaryType	ISO8601DateTime	UPDMElement	\$OutOfServiceMilestone[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$OutOfServiceMilestone[i].propertySet
resource	SystemResource	ActualProjectMilestone	\$OutOfServiceMilestone[i].resource
startBoundaryType	ISO8601DateTime	UPDMElement	\$OutOfServiceMilestone[i].startBoundaryType
URI	String	UPDMElement	\$OutOfServiceMilestone[i].URI

589. OV-1

Base Classifier

- [InvisibleStereotype](#)

590. OV-1 Package

Base Classifier

- [InvisibleStereotype](#)

591. OV-1i

Base Classifier

- [InvisibleStereotype](#)

592. OV-2

Base Classifier

- [InvisibleStereotype](#)

593. OV-2 Package

Base Classifier

- [InvisibleStereotype](#)

594. OV-2 Package MODAF

Base Classifier

- [InvisibleStereotype](#)

595. OV-2i

Base Classifier

- [InvisibleStereotype](#)

596. OV-3

The Operational Information Exchange Matrix (OV-3) addresses operational exchanges between nodes.

To fill in OV-3 table, Existing Operational Exchanges of any available kind such as Information Exchange, Materiel Exchange, Organizational Exchange, Energy Exchange and etc. have to be added to it.

*Click **Add New** button to create a new Operational Exchange between selected Nodes.*

*Click **Add Existing** button to select Operational Exchanges or Needlines. In case Needline is selected, all Operational Exchanges flowing via it will be added to the table.*

"Operational Exchange Identifier", "Operational Exchange Item Name, Producing and Consuming Operational Activities, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Operational Exchanges OV-2 product is recommended to use.

Rows (Operational Exchanges) can be removed from model or only from table, can be filtered according to represented Operational Exchange kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	OV-3	\$OV-3[i].hideColumns

597. OV-3 Package

Base Classifier

- [InvisibleStereotype](#)

598. OV-3 Report

Base Classifier

- [InvisibleStereotype](#)

599. OV-3 Role Based

The Role-based Operational Information Exchange Matrix (OV-3) addresses the resources exchanged between node roles and the relevant attributes of the exchanges

To fill in OV-3 table, Existing Operational Exchanges of any available kind such as Information Exchange, Materiel Exchange, Organizational Exchange, Energy Exchange, Geo Political Extent Exchange and etc. have to be added to it.

*Click **Add Existing** button to select Operational Exchanges, Needlines, or Service Channels. In case Needline or Service Channel is selected, all Operational Exchanges flowing via it will be added to the table.*

"Operational Exchange ID", "Operational Exchange Item", "Producing and Consuming Operational Activities", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Operational Exchanges OV-2 internal diagram is recommended to use.

Rows (Operational Exchanges) can be removed from model or only from table, can be filtered according to represented Operational Exchange kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	OV-3 Role Based	\$OV-3RoleBased[i].hideColumns

600. OV-3 Role Based Report

Base Classifier

- [InvisibleStereotype](#)

601. OV-3-DoDAF2

Operational Resource Flow Matrix (OV-3) addresses the resources exchanged and the relevant attributes of the exchanges

To fill in OV-3 table, Existing Operational Exchanges of any available kind such as Information Exchange, Materiel Exchange, Organizational Exchange, Energy Exchange, Geo Political Extent Exchange and etc. have to be added to it.

Click **Add New** button to create a new Operational Exchange between selected Performers.

Click **Add Existing** button to select Operational Exchanges or Needlines. In case Needline is selected, all Operational Exchanges flowing via it will be added to the table.

"Operational Exchange Identifier", "Operational Exchange Item Name, Producing and Consuming Operational Activities, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Operational Exchanges OV-2 model is recommended to use.

Rows (*Operational Exchanges*) can be removed from model or only from table, can be filtered according to represented *Operational Exchange* kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	OV-3-DoDAF2	\$OV-3-DoDAF2 [i].hideColumns

602. OV-3-DoDAF2 Role Based

The Role-based Operational Resource Flow Matrix (OV-3) addresses the resources exchanged between node roles and the relevant attributes of the exchanges

To fill in OV-3 table, Existing *Operational Exchanges* of any available kind such as *Information Exchange*, *Materiel Exchange*, *Organizational Exchange*, *Energy Exchange*, *Geo Political Extent Exchange* and etc. have to be added to it.

Click **Add Existing** button to select *Operational Exchanges*, *Needlines*, or *Service Channels*. In case *Needline* or *Service Channel* is selected, all *Operational Exchanges* flowing via it will be added to the table.

"*Operational Exchange ID*", "*Operational Exchange Item*", "*Producing and Consuming Operational Activities*", and wide range of measurement cells are allowed to edit in the table. For creation and modification of *Operational Exchanges* OV-2 internal diagram is recommended to use.

Rows (*Operational Exchanges*) can be removed from model or only from table, can be filtered according to represented *Operational Exchange* kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	OV-3-DoDAF2 Role Based	\$OV-3-DoDAF2RoleBased [i].hideColumns

603. OV-4

Base Classifier

- [InvisibleStereotype](#)

604. OV-4 Package**Base Classifier**

- [InvisibleStereotype](#)

605. OV-5**Base Classifier**

- [InvisibleStereotype](#)

606. OV-5 Package**Base Classifier**

- [InvisibleStereotype](#)

607. OV-5 Package MODAF**Base Classifier**

- [InvisibleStereotype](#)

608. OV-6a

The Operational Rules Model (OV-6a) specifies operational or business rules that are constraints on an enterprise, a mission, operation, business, or architecture.

There are two ways to fill this table.

- 1. Add new Operational Constraint. Click **Add New** button and select constrained Node, Operational Activity, Entity Item, Mission, Exchange Element, or Operational Exchange). Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. Add Existing Operational Constraints. Click **Add Existing** button and select Operational Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Operational Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	OV-6a	\$OV-6a [i].hideColumns

609. OV-6a Package

Base Classifier

- [InvisibleStereotype](#)

610. OV-6a Report

Base Classifier

- [InvisibleStereotype](#)

611. OV-6a-DoDAF2

The Operational Rules Model (OV-6a) specifies operational or business rules that are constraints on an enterprise, a mission, operation, business, or architecture.

There are two ways to fill this table.

*1. Add new Operational Constraint. Click **Add New** button and select constrained Performer, Operational Activity, Entity Item, Mission, Exchange Element, or Operational Exchange). Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*

*2. Add Existing Operational Constraints. Click **Add Existing** button and select Operational Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Operational Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	OV-6a-DoDAF2	\$OV-6a-DoDAF2[i].hideColumns

612. OV-6b

Base Classifier

- [InvisibleStereotype](#)

613. OV-6b Package

Base Classifier

- [InvisibleStereotype](#)

614. OV-6c

Base Classifier

- [InvisibleStereotype](#)

615. OV-6c Package

Base Classifier

- [InvisibleStereotype](#)

616. OV-7

Base Classifier

- [InvisibleStereotype](#)

617. OV-7 Package

Base Classifier

- [InvisibleStereotype](#)

618. OV-7 Package MODAF

Base Classifier

- [InvisibleStereotype](#)

619. Overlap

IDEAS: A couple of wholePart couples where the part in each couple is the same.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Overlap[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Overlap[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Overlap[i].endBoundaryType</code>
Overlap.client		Overlap	<code>\$Overlap[i].Overlap.client</code>
Overlap.supplier		Overlap	<code>\$Overlap[i].Overlap.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$Overlap[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Overlap[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Overlap[i].URI</code>

620. Overwrite

When the «overwrite» stereotype is applied to object nodes, a token arriving at a full object node replaces the ones already there (a full object node has as many tokens as allowed by its upper bound). This is typically used on an input pin with an upper bound of 1 to ensure that stale data is overridden at an input pin. For upper bounds greater than one, the token replaced is the one that would be the last to be selected according to the ordering kind for the node. For FIFO ordering, this is the most recently added token, for LIFO it is the least recently added token. A null token removes all the tokens already there. The number of tokens replaced is equal to the weight of the incoming edge, which defaults to 1. For object nodes that are the target of continuous flows, «overwrite» and «nobuffer» have the same effect. The stereotype does not override UML token offering semantics, just indicates what happens to the token when it is accepted. When the stereotype is not applied, the semantics is as in UML, specifically, tokens arriving at object nodes do not replace ones that are already there.

621. OwnerDisplayMode

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Compact tree	Enumeration Literal	OwnerDisplayMode	<code>\$OwnerDisplayMode[i].Compact tree</code>
Complete tree	Enumeration Literal	OwnerDisplayMode	<code>\$OwnerDisplayMode[i].Complete tree</code>
Full qualified name	Enumeration Literal	OwnerDisplayMode	<code>\$OwnerDisplayMode[i].Full qualified name</code>
Hidden	Enumeration Literal	OwnerDisplayMode	<code>\$OwnerDisplayMode[i].Hidden</code>

622. OwnsProcess

UPDM: Asserts that an ActualOrganizationalResource owns a Process.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$OwnsProcess[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$OwnsProcess[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OwnsProcess[i].endBoundaryType</code>
OwnsProcess.client		OwnsProcess	<code>\$OwnsProcess[i].OwnsProcess.client</code>
OwnsProcess.supplier		OwnsProcess	<code>\$OwnsProcess[i].OwnsProcess.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$OwnsProcess[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$OwnsProcess[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$OwnsProcess[i].URI</code>

623. P1

Base Classifier

- [InvisibleStereotype](#)

624. P1 Actual Quality Requirements

System Actual Quality Requirements Description (P1 Actual) depicts the Actual values of performance characteristics of a Resource. There are three ways to add a row in this table:

1. **Add new** measurable Resource; Click **Add New** button and select one or more System Resources that have at least one Measurement Set Defined (see P1 Typical). Specify values for each Measurement - directly in the table cells.

2. **Add existing Measures or measurable Resources.** Click **Add Existing** button and select *Existing Measurements or Resources*.
3. **Add missing Measurements.** Click “**Add the missing Measurements**” button to update table to include latest model changes.

Rows (*Actual Measurements*) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	P1 Actual Quality Requirements	\$P1ActualQualityRequirements[i].hideColumns

625. P1 Package

Base Classifier

- [InvisibleStereotype](#)

626. P1 Report

Base Classifier

- [InvisibleStereotype](#)

627. P1 Service Provision

Service Provision (P1) Matrix defines the relationships between the Resources and Service Interfaces. The Rows of this matrix are Service Interfaces and the Columns are System Resources.

To build the Matrix:

1. **Specify Rows scope** (*Service Interfaces*);
2. **Specify Columns scope** (*System Resources*);
3. **Click "Refresh" button.**

Cells here represents Services or Requests. If the resource provides service, Service Icon is displayed. If it requests (consumes) service, Request Icon is displayed.

By double clicking on the empty cell new Service will be created. By double clicking once more new Request will be created and the Service will be deleted. By double clicking third time cell will be cleared and the Request will be deleted.

Base Classifier

- [InvisibleStereotype](#)

628. P1 Systems to Systems Matrix

Systems to Systems Matrix (P1) shows how resources interact to each other. It is editable matrix where cells of the matrix represent Resource interactions and the headers represent System Resources.

Arrow can be on one side and on the other side. It depends on the direction of Resource interaction flow between System Resources. P1 matrix consists of as much rows and columns as there are System Resources in the matrix data source.

To build the Matrix:

- 1. Specify Rows scope (System Resources);*
- 2. Specify Columns scope (System Resources);*
- 3. Click "Refresh" button.*

By clicking on the cell context menu opens allowing to open new Resource Interaction creation wizard or delete existing relationships.

Base Classifier

- [InvisibleStereotype](#)

629. P1 Technology Forecast

The Technology Forecast (P1) defines the underlying current and expected supporting technologies and skills. Three major steps should be done to create the table:

- 1. Add Rows to the Table. There are two ways to add a row to this table:*
 - 1.1. Add new Resource as row Header. Click **Add New** button and select System Resource you want to create. Specify owner for selected Resource.*
 - 1.2. Add Existing Resource as row Header. Click **Add Existing** button and select one or more existing System Resources.*

2. **Add columns** to the table. Click "**Add/Remove forecast**" button. Specify time periods for the forecasting: select or create Time Line Package to store forecast dates; select forecast kind. According to the selected forecast kind, specify additional options needed (see Forecasting Period Dialog help for more information).

3. **Fill in the cells** with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select System Resources, Standards, Protocols or Competences to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected time period.

Rows (System Resources) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

630. P1 Typical Requirements

System Typical Quality Requirements Description (P1 Typical) is the and depicts the possible types of performance characteristics of a Resource. There are two ways to add a row in this table:

1. **Add new Measurement Set.** Click **Add New** button and select the owning element for Measurement Set. Specify Measurements to the Measurement Set and Resources to be Measured - straight in the table cells.

2. **Add existing Measurements Sets.** Click **Add Existing** button and select Existing Measurement Sets.

Rows (Measurement Sets) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	P1 Typical Requirements	\$P1TypicalRequirements[i].hideColumns

631. P2 Package

Base Classifier

- [InvisibleStereotype](#)

632. P2i

Base Classifier

- [InvisibleStereotype](#)
- [SV-1i](#)

633. P3

The Resource Connectivity (P3) specifies the characteristics of the data exchanged between Resources. To fill in P3 table, Existing Resource Interactions have to be added to it.

*Click **Add New** button to create a new Resource Interaction between selected System Resources.*

*Click **Add Existing** button and select Resource Interactions or Resource Interfaces. In case Resource Interface is selected, all Resource Interactions flowing via it will be added to the table.*

"Resource Interaction identifier". "Resource Interaction Item Name, Producing and Consuming Functions, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Resource Interactions P1 product is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	P3	\$P3[i].hideColumns

634. P3 Package

Base Classifier

- [InvisibleStereotype](#)

635. P3 Report

Base Classifier

- [InvisibleStereotype](#)

636. P3 Role Based

The Resource Role Interactions (P3) addresses specifies the characteristics of the data exchanged between resources.

To fill in P3 table, Existing Resource Interactions have to be added to it.

*Click **Add Existing** button to select Resource Interactions, Resource Interfaces, Resource Connectors, or Service Channels. In case Resource Interface, Resource Connector, or Service Channel is selected, all Resource Interactions flowing via it will be added to the table.*

"Interaction ID", "Resource Interaction Item", "Producing and Consuming Functions", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Resource Interactions P2 diagram is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be filtered according to the connector kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	P3 Role Based	\$P3RoleBased[i].hideColumns

637. P3 Role Based Report

Base Classifier

- [InvisibleStereotype](#)

638. P4

Base Classifier

- [InvisibleStereotype](#)

639. P4 Package

Base Classifier

- [InvisibleStereotype](#)

640. P5

Base Classifier

- [InvisibleStereotype](#)

641. P5 Package

Base Classifier

- [InvisibleStereotype](#)

642. P6

Base Classifier

- [InvisibleStereotype](#)

643. P6 Package

Base Classifier

- [InvisibleStereotype](#)

644. P7

Base Classifier

- [InvisibleStereotype](#)

645. P7 Package

Base Classifier

- [InvisibleStereotype](#)

646. P8

Resource Constraints (P8) allows you to constraint Systems View Architectural elements. There are two ways to fill this table:

- 1. Add new Resource Constraint. Click **Add New** button and select Systems Element (Resource Artifact, Software, Capability Configuration, Organization, Post, Function, Exchange Element, Entity Item, Resource Interaction) to be constrained. Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. Add Existing Resource Constraints. Click **Add Existing** button and select Resource Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Resource Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

hideColumns	String	P8	\$P8[i].hideColumns
-------------	--------	----	---------------------

647. P8 Package

Base Classifier

- [InvisibleStereotype](#)

648. P8 Report

Base Classifier

- [InvisibleStereotype](#)

649. Panel

650. Paragraph Kind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Formal	Enumeration Literal	Paragraph Kind	\$ParagraphKind[i].Formal
Ordinary	Enumeration Literal	Paragraph Kind	\$ParagraphKind[i].Ordinary
Simple	Enumeration Literal	Paragraph Kind	\$ParagraphKind[i].Simple

651. ParameterDefinition

Metric property. Contains value to be used as parameter in other metrics

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
abbreviation	String	ParameterDefinition	\$ParameterDefinition[i].abbreviation
setAsContext	boolean	ParameterDefinition	\$ParameterDefinition[i].setAsContext

652. ParameterDirectionKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
in	Enumeration Literal	ParameterDirectionKind	<code>\$ParameterDirectionKind[i].in</code>
inout	Enumeration Literal	ParameterDirectionKind	<code>\$ParameterDirectionKind[i].inout</code>
out	Enumeration Literal	ParameterDirectionKind	<code>\$ParameterDirectionKind[i].out</code>
return	Enumeration Literal	ParameterDirectionKind	<code>\$ParameterDirectionKind[i].return</code>

653. ParameterEffectKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
create	Enumeration Literal	ParameterEffectKind	<code>\$ParameterEffectKind[i].create</code>
delete	Enumeration Literal	ParameterEffectKind	<code>\$ParameterEffectKind[i].delete</code>
read	Enumeration Literal	ParameterEffectKind	<code>\$ParameterEffectKind[i].read</code>
update	Enumeration Literal	ParameterEffectKind	<code>\$ParameterEffectKind[i].update</code>

654. Participant

UPDM: A participant is the abstract type of a provider and/or consumer of services. In the business domain a participant may be a person, organization or system. In the systems domain a participant may be a system, application or component.

Base Classifier

- [CapableElement](#)
- [ConceptItem](#)
- [Desirer](#)
- [OperationalExchangeItem](#)
- [Participant](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$Participant[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Participant[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Participant[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Participant[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Participant[i].endBoundaryType</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$Participant[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$Participant[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$Participant[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Participant[i].startBoundaryType</code>

URI	String	UPDMElement	\$Participant[i].URI
-----	--------	-----------------------------	----------------------

655. Participant

A participant is the type of a provider and/or consumer of services. In the business domain a participant may be a person, organization or system. In the systems domain a participant may be a system, application or component.

656. ParticipantProperty

The Block stereotype extends Class, so it can be applied to any specialization of Class, including Association Classes. These are informally called “association blocks.” An association block can own properties and connectors, like any other block. Each instance of an association block can link together instances of the end classifiers of the association. To refer to linked objects and values of an instance of an association block, it is necessary for the modeler to specify which (participant) properties of the association block identify the instances being linked at which end of the association. The value of a participant property on an instance (link) of the association block is the value or object at the end of the link corresponding to this end of the association.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
end	Property	ParticipantProperty	\$ParticipantProperty[i].end

657. PartProperty

A part property specifies part, with stronger ownership and coincidental lifetime, of its containing block. Every part property has 'composite' AggregationKind and is typed by a block. A part property will be displayed under the 'parts' compartment when 'Sort by SysML Style' in the 'Presentation Options' is selected. A part property describes a local usage or role of the typing block in the context of the containing block.

Base Classifier

- [BlockProperty](#)

658. Pattern

659. performanceRequirement

Requirement that quantitatively measures the extent to which a system, or a system part, satisfies a required capability or condition.

Base Classifier

- [extendedRequirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	\$performanceRequirement[i].Derived

DerivedFrom	Requirement	Requirement	<code>\$performanceRequirement[i].DerivedFrom</code>
Id	String	Requirement	<code>\$performanceRequirement[i].Id</code>
Master	Requirement	Requirement	<code>\$performanceRequirement[i].Master</code>
RefinedBy	NamedElement	Requirement	<code>\$performanceRequirement[i].RefinedBy</code>
risk	RiskKind	extendedRequirement	<code>\$performanceRequirement[i].risk</code>
SatisfiedBy	NamedElement	Requirement	<code>\$performanceRequirement[i].SatisfiedBy</code>
source	String	extendedRequirement	<code>\$performanceRequirement[i].source</code>
Text	String	Requirement	<code>\$performanceRequirement[i].Text</code>
TracedTo	NamedElement	Requirement	<code>\$performanceRequirement[i].TracedTo</code>
VerifiedBy	NamedElement	Requirement	<code>\$performanceRequirement[i].VerifiedBy</code>
verifyMethod	VerificationMethodKind	extendedRequirement	<code>\$performanceRequirement[i].verifyMethod</code>

660. Performer

MODAF:NA

DoDAF: Any entity - human, automated, or any aggregation of human and/or automated - that performs an activity and provides a capability.

Base Classifier

- [Node](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$Performer[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Performer[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Performer[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Performer[i].conformsTo</code>
connectedNodes	Node	Node	<code>\$Performer[i].connectedNodes</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Performer[i].endBoundaryType</code>
Node.ownedPort		Node	<code>\$Performer[i].Node.ownedPort</code>
Node.performs		Node	<code>\$Performer[i].Node.performs</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$Performer[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$Performer[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$Performer[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Performer[i].startBoundaryType</code>
SubjectOfOperationalStateMachine.ownedBehavior		SubjectOfOperationalStateMachine	<code>\$Performer[i].SubjectOfOperationalStateMachine.ownedBehavior</code>
URI	String	UPDMElement	<code>\$Performer[i].URI</code>

661. Person

UPDM: A type of a human being that is recognized by law as the subject of rights and duties. This is used to define the characteristics that require capturing for ActualPersons (e.g. properties such as address, rank, telephone number, etc).

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Person[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Person[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Person[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Person[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Person[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Person[i].URI</code>

662. PersonType

DoDAF: A category of persons defined by the role or roles they share that are relevant to an architecture. Includes assigned materiel.

MODAF: NA

Base Classifier

- [Post](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$PersonType[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$PersonType[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$PersonType[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$PersonType[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$PersonType[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PersonType[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$PersonType[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$PersonType[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$PersonType[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$PersonType[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$PersonType[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$PersonType[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$PersonType[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PersonType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$PersonType[i].URI</code>

663. Physical Resource Viewpoint

Base Classifier

- [InvisibleStereotype](#)

664. PhysicalArchitecture

MODAF: A configuration of Resources for a purpose.

DoDAF: NA

Base Classifier

- [SystemResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$PhysicalArchitecture[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$PhysicalArchitecture[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$PhysicalArchitecture[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$PhysicalArchitecture[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$PhysicalArchitecture[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PhysicalArchitecture[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$PhysicalArchitecture[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$PhysicalArchitecture[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$PhysicalArchitecture[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$PhysicalArchitecture[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$PhysicalArchitecture[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$PhysicalArchitecture[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$PhysicalArchitecture[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PhysicalArchitecture[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$PhysicalArchitecture[i].URI</code>

665. PhysicalDataModel

MODAF: A PhysicalDataModel is an implementable specification of a data structure. A PhysicalDataModel realises a LogicalDataModel, taking into account implementation restrictions and performance issues whilst still enforcing the constraints, relationships and typing of the logical model.

DoDAF: A Physical Data Model defines the structure of the various kinds of system or service data that are utilized by the systems or services in the Architecture.

Base Classifier

- [DataModel](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$PhysicalDataModel[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$PhysicalDataModel[i].conformsTo</code>
DataModel.ownedElement		DataModel	<code>\$PhysicalDataModel[i].DataModel.ownedElement</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PhysicalDataModel[i].endBoundaryType</code>
physicalDataModelType	String	PhysicalDataModel	<code>\$PhysicalDataModel[i].physicalDataModelType</code>
propertySet	PropertySet	UPDMElement	<code>\$PhysicalDataModel[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PhysicalDataModel[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$PhysicalDataModel[i].URI</code>

666. physicalRequirement

Requirement that specifies physical characteristics and/or physical constraints of the system, or a system part.

Base Classifier

- [extendedRequirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	<code>\$physicalRequirement[i].Derived</code>
DerivedFrom	Requirement	Requirement	<code>\$physicalRequirement[i].DerivedFrom</code>
Id	String	Requirement	<code>\$physicalRequirement[i].Id</code>
Master	Requirement	Requirement	<code>\$physicalRequirement[i].Master</code>
RefinedBy	NamedElement	Requirement	<code>\$physicalRequirement[i].RefinedBy</code>
risk	RiskKind	extendedRequirement	<code>\$physicalRequirement[i].risk</code>
SatisfiedBy	NamedElement	Requirement	<code>\$physicalRequirement[i].SatisfiedBy</code>
source	String	extendedRequirement	<code>\$physicalRequirement[i].source</code>
Text	String	Requirement	<code>\$physicalRequirement[i].Text</code>
TracedTo	NamedElement	Requirement	<code>\$physicalRequirement[i].TracedTo</code>
VerifiedBy	NamedElement	Requirement	<code>\$physicalRequirement[i].VerifiedBy</code>
verifyMethod	VerificationMethodKind	extendedRequirement	<code>\$physicalRequirement[i].verifyMethod</code>

667. PhysicalResource

Base Classifier

- [SystemResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$PhysicalResource[i].actsUpon</code>

actualPropertySet	ActualPropertySet	UPDMElement	<code>\$PhysicalResource[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$PhysicalResource[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$PhysicalResource[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$PhysicalResource[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PhysicalResource[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$PhysicalResource[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$PhysicalResource[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$PhysicalResource[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$PhysicalResource[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$PhysicalResource[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$PhysicalResource[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$PhysicalResource[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PhysicalResource[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$PhysicalResource[i].URI</code>

668. placeOnPaletteProperty

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
place	String	placeOnPaletteProperty	<code>\$placeOnPaletteProperty[i].place</code>

669. Port

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
connectorRequired	Boolean	Port	<code>\$Port[i].connectorRequired</code>

670. Post

MODAF: A Post (MODAF::PostType) is a type of point of contact or responsible person. Note that this is the type of post - e.g. Desk Officer, Commander Land Component, etc.

DoDAF: A Post (DoDAF::PersonType) is a category of persons defined by the role or roles they share that are relevant to an architecture.

Base Classifier

- [CompetenceRequirer](#)
- [OrganizationalResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$Post[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Post[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$Post[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$Post[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Post[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Post[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$Post[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$Post[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$Post[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$Post[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$Post[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$Post[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$Post[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Post[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Post[i].URI</code>

671. PostConceptRole

Base Classifier

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$PostConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$PostConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$PostConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PostConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$PostConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PostConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$PostConceptRole[i].URI</code>

672. Pr

Base Classifier

- [InvisibleStereotype](#)
- [SV-8](#)

673. Pr Package

Base Classifier

- [InvisibleStereotype](#)

674. Pr Report

Base Classifier

- [InvisibleStereotype](#)

675. Probability

When the «probability» stereotype is applied to edges coming out of decision nodes and object nodes, it provides an expression for the probability that the edge will be traversed. These must be between zero and one inclusive, and add up to one for edges with same source at the time the probabilities are used.

When the «probability» stereotype is applied to output parameter sets, it gives the probability the parameter set will be given values at runtime. These must be between zero and one inclusive, and add up to one for output parameter sets of the same behavior at the time the probabilities are used.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
probability	String	Probability	\$Probability[i].probability

676. Problem

A *Problem* documents a deficiency, limitation, or failure of one or more model elements to satisfy a requirement or need, or other undesired outcome. It may be used to capture problems identified during analysis, design, verification, or manufacture and associate the problem with the relevant model elements. *Problem* is a stereotype of comment and may be attached to any other model element in the same manner as a comment.

677. ProblemDomain

MODAF: The boundary containing those Nodes which may be realised by functional resources specified in SV-1. There may be more than one alternative solution for a given *ProblemDomain* specified as a set of SV suites. There may be only one *ProblemDomain* in a *LogicalArchitecture*.

DoDAF: NA – covered by the more general *temporalWholePart* element.

Base Classifier

- [NodeRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProblemDomain[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProblemDomain[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProblemDomain[i].endBoundaryType</code>
NodeRole.class		NodeRole	<code>\$ProblemDomain[i].NodeRole.class</code>
NodeRole.type		NodeRole	<code>\$ProblemDomain[i].NodeRole.type</code>
performsInContext	OperationalActivity	NodeRole	<code>\$ProblemDomain[i].performsInContext</code>
ProblemDomain.class		ProblemDomain	<code>\$ProblemDomain[i].ProblemDomain.class</code>
ProblemDomain.type		ProblemDomain	<code>\$ProblemDomain[i].ProblemDomain.type</code>
propertySet	PropertySet	UPDMElement	<code>\$ProblemDomain[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProblemDomain[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ProblemDomain[i].URI</code>

678. Process

A transaction based component.

679. Process

MODAF: The abstract supertype of OperationalActivity and EnduringTask.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Process[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Process[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Process[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Process[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Process[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Process[i].URI</code>

680. processView

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
processViewID	int	processView	<code>\$processView[i].processViewID</code>

681. ProfileUpgradeMappingRule

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
		ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].</code>
addToDolfTagDoesNotExist	boolean	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].addToDoIfTagDoesNotExist</code>
caseSensitiveEnumerationLiteral	boolean	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].caseSensitiveEnumerationLiteral</code>
disableNewTypeCreation	boolean	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].disableNewTypeCreation</code>
disableReplaceWhereSavedAsElementValue	boolean	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].disableReplaceWhereSavedAsElementValue</code>
explicitNewMetaclass	Element	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].explicitNewMetaclass</code>
searchForDerivedIfDoesNotExist	boolean	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].searchForDerivedIfDoesNotExist</code>
sourceStereotypeIconContent	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].sourceStereotypeIconContent</code>
sourceStereotypeIconFormat	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].sourceStereotypeIconFormat</code>
sourceStereotypeID	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].sourceStereotypeID</code>
sourceStereotypeMetaclass	Element	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].sourceStereotypeMetaclass</code>
sourceStereotypeName	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].sourceStereotypeName</code>
sourceTagID	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].sourceTagID</code>
sourceTagName	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].sourceTagName</code>
targetStereotypeIconContent	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].targetStereotypeIconContent</code>
targetStereotypeIconFormat	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].targetStereotypeIconFormat</code>
targetStereotypeID	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].targetStereotypeID</code>
targetStereotypeName	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].targetStereotypeName</code>
targetTagID	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].targetTagID</code>
targetTagName	String	ProfileUpgradeMappingRule	<code>\$ProfileUpgradeMappingRule[i].targetTagName</code>

682. ProfileUpgradeTable

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
		ProfileUpgradeTable	<code>\$ProfileUpgradeTable[i].</code>
sourceProjectName	String	ProfileUpgradeTable	<code>\$ProfileUpgradeTable[i].sourceProjectName</code>
targetProjectName	String	ProfileUpgradeTable	<code>\$ProfileUpgradeTable[i].targetProjectName</code>

683. Project

DoDAF: A temporary endeavor undertaken to create Resources or Desired Effects.

Base Classifier

- [ActualProject](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActualProject.classifier		ActualProject	<code>\$Project[i].ActualProject.classifier</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Project[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Project[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Project[i].endBoundaryType</code>
endDate	ISO8601DateTime	ActualProject	<code>\$Project[i].endDate</code>
ownedMilestones	ActualProjectMilestone	ActualProject	<code>\$Project[i].ownedMilestones</code>
part	ActualProject	ActualProject	<code>\$Project[i].part</code>
propertySet	PropertySet	UPDMElement	<code>\$Project[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Project[i].startBoundaryType</code>
startDate	ISO8601DateTime	ActualProject	<code>\$Project[i].startDate</code>
URI	String	UPDMElement	<code>\$Project[i].URI</code>
whole	ActualProject	ActualProject	<code>\$Project[i].whole</code>

684. Project Sequence Types

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Finish-to-Finish	Enumeration Literal	Project Sequence Types	<code>\$ProjectSequenceTypes[i].Finish-to-Finish</code>
Finish-to-Start	Enumeration Literal	Project Sequence Types	<code>\$ProjectSequenceTypes[i].Finish-to-Start</code>

Start-to-Finish	Enumeration Literal	Project Sequence Types	<code>\$ProjectSequenceTypes[i].Start-to-Finish</code>
Start-to-Start	Enumeration Literal	Project Sequence Types	<code>\$ProjectSequenceTypes[i].Start-to-Start</code>

685. Project Viewpoint

Base Classifier

- [InvisibleStereotype](#)

686. ProjectActivity

MOAF: NA

DoDAF: An activity carried out during a project.

Base Classifier

- [Activity](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
activityPerformableUnderCondition	Environment	Activity	<code>\$ProjectActivity[i].activityPerformableUnderCondition</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProjectActivity[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProjectActivity[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectActivity[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ProjectActivity[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectActivity[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ProjectActivity[i].URI</code>

687. ProjectActivityAction

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProjectActivityAction[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProjectActivityAction[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectActivityAction[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ProjectActivityAction[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectActivityAction[i].startBoundaryType</code>

URI	String	UPDMElement	\$ProjectActivityAction[i].URI
-----	--------	-----------------------------	--------------------------------

688. ProjectActivityEdge

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ProjectActivityEdge[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ProjectActivityEdge[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectActivityEdge[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ProjectActivityEdge[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectActivityEdge[i].startBoundaryType
URI	String	UPDMElement	\$ProjectActivityEdge[i].URI

689. ProjectMilestone

UPDM: An element representing a collection of themes (e.g. DLOD or DOTMLPF) which is connected to a Project as part of a Project's definition. This is used as a template for ActualProjectMilestones.

MODAF: An event in a Project by which progress is measured.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ProjectMilestone[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ProjectMilestone[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectMilestone[i].endBoundaryType
ProjectMilestone.ownedAttributes		ProjectMilestone	\$ProjectMilestone[i].ProjectMilestone.ownedAttributes
ProjectMilestone.ownedThemes		ProjectMilestone	\$ProjectMilestone[i].ProjectMilestone.ownedThemes
propertySet	PropertySet	UPDMElement	\$ProjectMilestone[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectMilestone[i].startBoundaryType
URI	String	UPDMElement	\$ProjectMilestone[i].URI

690. ProjectMilestoneRole

UPDM: An instance of a ProjectMilestoneRole in the context of an ActualProject.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ProjectMilestoneRole[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ProjectMilestoneRole[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectMilestoneRole[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ProjectMilestoneRole[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectMilestoneRole[i].startBoundaryType
URI	String	UPDMElement	\$ProjectMilestoneRole[i].URI

691. ProjectOwnership

MODAF: A type of OrganisationProjectRelationship where the organisation is the party responsible for the project.

Base Classifier

- [OrganizationalProjectRelationship](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ProjectOwnership[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ProjectOwnership[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectOwnership[i].endBoundaryType
endDate	ISO8601DateTime	OrganizationalProjectRelationship	\$ProjectOwnership[i].endDate
OrganizationalProjectRelationship.client		OrganizationalProjectRelationship	\$ProjectOwnership[i].OrganizationalProjectRelationship.client
OrganizationalProjectRelationship.supplier		OrganizationalProjectRelationship	\$ProjectOwnership[i].OrganizationalProjectRelationship.supplier
propertySet	PropertySet	UPDMElement	\$ProjectOwnership[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$ProjectOwnership[i].startBoundaryType
startDate	ISO8601DateTime	OrganizationalProjectRelationship	\$ProjectOwnership[i].startDate
URI	String	UPDMElement	\$ProjectOwnership[i].URI

692. ProjectSequence

MODAF: Asserts that one ActualProject (MODAF::Project) follows from another - i.e. the target ActualProject cannot start until the source ActualProject has ended.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProjectSequence[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProjectSequence[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectSequence[i].endBoundaryType</code>
ProjectSequence.client		ProjectSequence	<code>\$ProjectSequence[i].ProjectSequence.client</code>
ProjectSequence.supplier		ProjectSequence	<code>\$ProjectSequence[i].ProjectSequence.supplier</code>
propertySet	PropertySet	UPDMElement	<code>\$ProjectSequence[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectSequence[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ProjectSequence[i].URI</code>

693. ProjectStatus

MODAF: A ProjectStatus (MODAF::StatusAtMilestone) is a relationship between a Status and a milestone that asserts the status (i.e. level of progress) of a ProjectTheme for the project at the time of the ActualProjectMilestone (MODAF::Milestone).

DoDAF: NA

Base Classifier

- [ActualProperty](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProjectStatus[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProjectStatus[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectStatus[i].endBoundaryType</code>
endDate	ISO8601DateTime	ActualProperty	<code>\$ProjectStatus[i].endDate</code>
intention	ActualPropertySetKind	ActualProperty	<code>\$ProjectStatus[i].intention</code>
ProjectStatus.definingFeature		ProjectStatus	<code>\$ProjectStatus[i].ProjectStatus.definingFeature</code>
propertySet	PropertySet	UPDMElement	<code>\$ProjectStatus[i].propertySet</code>
PropertyValue.definingFeature		ActualProperty	<code>\$ProjectStatus[i].PropertyValue.definingFeature</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectStatus[i].startBoundaryType</code>
startDate	ISO8601DateTime	ActualProperty	<code>\$ProjectStatus[i].startDate</code>
URI	String	UPDMElement	<code>\$ProjectStatus[i].URI</code>

694. ProjectTheme

MODAF: An aspect by which the progress of various Projects may be measured. In UK MOD, this could be one of the defence lines of development (DLOD), or DOTMLPF in the US.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProjectTheme[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProjectTheme[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectTheme[i].endBoundaryType</code>
ProjectTheme.type		ProjectTheme	<code>\$ProjectTheme[i].ProjectTheme.type</code>
propertySet	PropertySet	UPDMElement	<code>\$ProjectTheme[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectTheme[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ProjectTheme[i].URI</code>

695. ProjectType

MODAF: A Project (MODAF::ProjectType) is used to define a category of project: For example, "Programme", "Acquisition Project" or "Training Programme".

DoDAF: NA (only Individual Project in DoDAF).

Base Classifier

- [Desirer](#)
- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProjectType[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProjectType[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectType[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ProjectType[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProjectType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ProjectType[i].URI</code>

696. Property

The Property stereotype augments the standard UML Property with the ability to be distinguished as an identifying property meaning the property can be used to distinguish instances of the containing Classifier. This is also known as a "primary key". In the context of SoaML the ID is used to distinguish the correlation identifier in a message.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isID	Boolean	Property	<code>\$Property[i].isID</code>

697. Property

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Property[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Property[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Property[i].endBoundaryType</code>
maxValue	String	Property	<code>\$Property[i].maxValue</code>
minValue	String	Property	<code>\$Property[i].minValue</code>
propertySet	PropertySet	UPDMElement	<code>\$Property[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Property[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Property[i].URI</code>

698. propertyGroup

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
columns	String	propertyGroup	<code>\$propertyGroup[i].columns</code>
filter	Class	propertyGroup	<code>\$propertyGroup[i].filter</code>
properties	String	propertyGroup	<code>\$propertyGroup[i].properties</code>
showGroupInCompartmentEdit	boolean	propertyGroup	<code>\$propertyGroup[i].showGroupInCompartmentEdit</code>
showGroupInDependencyMatrix	boolean	propertyGroup	<code>\$propertyGroup[i].showGroupInDependencyMatrix</code>
showGroupInElementSpecification	boolean	propertyGroup	<code>\$propertyGroup[i].showGroupInElementSpecification</code>
showGroupInGoTo	boolean	propertyGroup	<code>\$propertyGroup[i].showGroupInGoTo</code>
showGroupInQuickProperties	boolean	propertyGroup	<code>\$propertyGroup[i].showGroupInQuickProperties</code>
showGroupInRelationMap	boolean	propertyGroup	<code>\$propertyGroup[i].showGroupInRelationMap</code>
titleBarDescription	String	propertyGroup	<code>\$propertyGroup[i].titleBarDescription</code>
titleBarName	String	propertyGroup	<code>\$propertyGroup[i].titleBarName</code>
useAsNode	Boolean	propertyGroup	<code>\$propertyGroup[i].useAsNode</code>

699. PropertySet

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$PropertySet[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$PropertySet[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$PropertySet[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PropertySet[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$PropertySet[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$PropertySet[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$PropertySet[i].URI</code>

700. PropertySpecificType

The PropertySpecificType stereotype should automatically be applied to the classifier which types a property with a property-specific type. This classifier can contain definitions of new or redefined features which extend the original classifier referenced by the property-specific type.

701. ProprietaryInformation

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
not known	Enumeration Literal	ProprietaryInformation	<code>\$ProprietaryInformation[i].not known</code>
not proprietary information	Enumeration Literal	ProprietaryInformation	<code>\$ProprietaryInformation[i].not proprietary information</code>
not specified	Enumeration Literal	ProprietaryInformation	<code>\$ProprietaryInformation[i].not specified</code>
proprietary information	Enumeration Literal	ProprietaryInformation	<code>\$ProprietaryInformation[i].proprietary information</code>

702. Protocol

MODAF: A Standard for communication. Protocols may be composite (i.e. a stack).

DoDAF: NA, See TechnicalStandard.

Base Classifier

- [Standard](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Protocol[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Protocol[i].conformsTo</code>
currentStatus	String	Standard	<code>\$Protocol[i].currentStatus</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Protocol[i].endBoundaryType</code>
InformationTechnologyStandardCategory	String	Standard	<code>\$Protocol[i].InformationTechnologyStandardCategory</code>
mandatedDate	ISO8601DateTime	Standard	<code>\$Protocol[i].mandatedDate</code>
propertySet	PropertySet	UPDMElement	<code>\$Protocol[i].propertySet</code>
ratifiedBy	ActualOrganization	Standard	<code>\$Protocol[i].ratifiedBy</code>
retiredDate	ISO8601DateTime	Standard	<code>\$Protocol[i].retiredDate</code>
shortName	String	Standard	<code>\$Protocol[i].shortName</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Protocol[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Protocol[i].URI</code>
version	String	Standard	<code>\$Protocol[i].version</code>

703. ProtocolImplementation

UPDM: Abstract element: A connector that implements a specific Protocol.

MODAF: An element that can implement a Protocol.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProtocolImplementation[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProtocolImplementation[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProtocolImplementation[i].endBoundaryType</code>
implements	Protocol	ProtocolImplementation	<code>\$ProtocolImplementation[i].implements</code>
propertySet	PropertySet	UPDMElement	<code>\$ProtocolImplementation[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProtocolImplementation[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ProtocolImplementation[i].URI</code>

704. ProtocolLayer

MODAF: Asserts that a Protocol (upperLayer) uses another Protocol (lowerLayer)

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProtocolLayer[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProtocolLayer[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProtocolLayer[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ProtocolLayer[i].propertySet</code>
ProtocolLayer.class		ProtocolLayer	<code>\$ProtocolLayer[i].ProtocolLayer.class</code>
ProtocolLayer.type		ProtocolLayer	<code>\$ProtocolLayer[i].ProtocolLayer.type</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProtocolLayer[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ProtocolLayer[i].URI</code>

705. Provider

Defines Interface for provider role in Service Contact.

706. ProvidesCompetence

UPDM: Asserts that a Resource type provides a competence.

MODAF: Asserts that a Role requires a Competence (MODAF::CompetenceForRole).

DoDAF: An overlap between a Personnel Type and the Skills it entails (DoDAF::skillPartOfPersonType)

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ProvidesCompetence[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ProvidesCompetence[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProvidesCompetence[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ProvidesCompetence[i].propertySet</code>
ProvidesCompetence.client		ProvidesCompetence	<code>\$ProvidesCompetence[i].ProvidesCompetence.client</code>
ProvidesCompetence.supplier		ProvidesCompetence	<code>\$ProvidesCompetence[i].ProvidesCompetence.supplier</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ProvidesCompetence[i].startBoundaryType</code>
universalPropertySet	ActualPropertySet	ProvidesCompetence	<code>\$ProvidesCompetence[i].universalPropertySet</code>
URI	String	UPDMElement	<code>\$ProvidesCompetence[i].URI</code>

707. ProxyPort

708. PseudostateKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
choice	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].choice</code>
deepHistory	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].deepHistory</code>
entryPoint	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].entryPoint</code>
exitPoint	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].exitPoint</code>
fork	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].fork</code>
initial	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].initial</code>
join	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].join</code>
junction	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].junction</code>
shallowHistory	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].shallowHistory</code>
terminate	Enumeration Literal	PseudostateKind	<code>\$PseudostateKind[i].terminate</code>

709. PV-1

The Responsibility Matrix (PV-1) describes the mapping between the Actual Projects and the Actual Organizational Resources.

The Rows of this matrix are Actual Projects and the Columns are Actual Organizational Resources (Actual Organization or Actual Post).

To build the Matrix:

- 1. Specify Rows scope (Actual Projects);*
- 2. Specify Columns scope (Actual Organizations and Actual Posts);*
- 3. Click "Refresh" button.*

Actual Organizational Resources maps to Actual Projects by "responsibleFor" property.

To map Actual Project to Actual Organizational Resource, Click on the intersection between the desired elements. By pressing on the intersection one more time, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

710. PV-1 Package

Base Classifier

- [InvisibleStereotype](#)

711. PV-2**Base Classifier**

- [InvisibleStereotype](#)

712. PV-2 Package**Base Classifier**

- [InvisibleStereotype](#)

713. PV-3

Project to Capability Mapping (PV-3) describes the mapping of programs and projects to capabilities to show how the specific projects and program elements help to achieve a capability.

The Rows of this matrix are Capabilities and the Columns are Projects.

To build the Matrix:

- 1. Specify Rows scope (Capabilities);*
- 2. Specify Columns scope (Projects);*
- 3. Click "Refresh" button.*

Capabilities are mapped with programs and projects using Resources and Activities. Resource realizes Capability using the Capability of Performer relationship and is assigned to the Milestone which is owned by the Actual Project. Activity realizes Capability using Activity Part of Capability relationship and is associated with an Actual Project using Activity Part of Project relationship.

Base Classifier

- [InvisibleStereotype](#)

714. PV-3 Package

Base Classifier

- [InvisibleStereotype](#)

715. QuantityKind

A *QuantityKind* is a kind of quantity that may be stated by means of defined units. For example, the quantity kind of length may be measured by units of meters, kilometers, or feet. *QuantityKind* is defined as a stereotype of *InstanceSpecification*, but it uses this metaclass only to define supporting elements for *ValueType* definitions. (The reuse of *InstanceSpecification* to define another metaclass is similar to the *EnumerationLiteral* metaclass in UML.) The only valid use of a *QuantityKind* instance is to be referenced by the “quantityKind” property of a *ValueType* or *Unit* stereotype.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
definitionURI	String	QuantityKind	\$QuantityKind[i].definitionURI
description	String	QuantityKind	\$QuantityKind[i].description
symbol	String	QuantityKind	\$QuantityKind[i].symbol

716. Rate

When the «rate» stereotype is applied to an activity edge, it specifies the expected value of the number of objects and values that traverse the edge per time interval, that is, the expected value rate at which they leave the source node and arrive at the target node. It does not refer to the rate at which a value changes over time. When the stereotype is applied to a parameter, the parameter must be streaming, and the stereotype gives the number of objects or values that flow in or out of the parameter per time interval while the behavior or operation is executing. Streaming is a characteristic of UML behavior parameters that supports the input and output of items while a behavior is executing, rather than only when the behavior starts and stops. The flow may be continuous or discrete. The «rate» stereotype has a rate property of type *InstanceSpecification*. The values of this property must be instances of classifiers stereotyped by «valueType» or «distributionDefinition». In particular, the denominator for units used in the rate property must be time units.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
rate	InstanceSpecification	Rate	\$Rate[i].rate

717. Rationale

A *Rationale* documents the justification for decisions and the requirements, design, and other decisions. A *Rationale* can be attached to any model element including relationships. It allows the user, for example, to specify a rationale that may reference more detailed documentation such as a trade study or analysis report. *Rationale* is a stereotype of comment and may be attached to any other model element in the same manner as a comment.

718. Real

A Real value type represents the mathematical concept of a real number. A Real value type may be used to type values that hold continuous quantities, without committing a specific representation such as a floating point data type with restrictions on precision and scale.

Base Classifier

- [Number](#)

719. Realization

A classifier that specifies a domain of objects and that also defines the physical implementation of those objects. For example, a Component stereotyped by «realization» will only have realizing Classifiers that implement behavior specified by a separate «specification» Component. See «specification». This differs from «implementation class» because an «implementation class» is a realization of a Class which can have features such as attributes and methods which is useful to system designers.

720. RealizingElement

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
realizedInformationFlows	Element	RealizingElement	\$RealizingElement[i].realizedInformationFlows

721. ReferenceProperty

A reference property specifies a reference of its containing block to another block. Every reference property has 'none' AggregationKind and is typed by a block. A reference property will be displayed under the 'references' compartment when 'Sort by SysML Style' in the 'Presentation Options' is selected.

Base Classifier

- [AbstractReferenceProperty](#)

722. Refine

Base Classifier

- [DirectedRelationshipPropertyPath](#)
- [Refine](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getRefines	Requirement	Refine	<code>\$Refine[i].getRefines</code>
sourceContext	Classifier	DirectedRelationshipPropertyPath	<code>\$Refine[i].sourceContext</code>
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$Refine[i].sourcePropertyPath</code>
targetContext	Classifier	DirectedRelationshipPropertyPath	<code>\$Refine[i].targetContext</code>
targetPropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$Refine[i].targetPropertyPath</code>

723. Refine

Specifies a refinement relationship between model elements at different semantic levels, such as analysis and design. The mapping specifies the relationship between the two elements or sets of elements. The mapping may or may not be computable, and it may be unidirectional or bidirectional. Refinement can be used to model transformations from analysis to design and other such changes.

724. RelationMap

*The **Relation Map** allows rapid creation, review, and analysis of relationships among the elements of the whole model.*

To create a relation map:

1. *Drag an element from the Model Browser to the relation map.*
2. *Specify filters: Relation Criterion, Element Type, Scope, and Depth.*

*Relation map can display the model structure in two different layouts - **tree** or **radial**.*

The relations represented on the map can be analyzed in one of the following ways:

Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

Statically - step-by-step by expanding the branches of the structure.

*To represent the same element as a single node in the map, click the Options button on the toolbar and then select the **Show Single Node Per Element** command.*

Manipulations:


Create a relation map structure - drag an element from the Model Browser to the relation map.

Restore manually suppressed / expanded branches and hidden elements and they position - click the Refresh button .

Expand / suppress branches - click  /  near the node.

Move the whole structure - click an empty space on the relation map and drag.

Move the selected node - click the node and drag.

Insert New Element - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.

Insert New Element of the Same Type - type the name and press Ctrl+Enter to create an element of the same type.

Zoom in - hold down Ctrl while scrolling up.

Zoom out - hold down Ctrl while scrolling down.

Fit in window - press Ctrl+W.

Zoom 1:1 - press Ctrl+NumPad /.

*Turn on the ability to change the context element with single click - click the Options button on the toolbar and then select the **Make Element as Context on Selection** command.*

Filters:

Context - the core element of the structure.

Relation Criterion - relations that are shown. Property, custom Tag definition, or relationship from diagrams can be chosen as a relation.

Element Type - element types that are shown.

Scope - fragment of the model (or the whole model) from which the relation map is built.

Depth - level of branches automatically expanded starting from the context element.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
contextElement	Element	RelationMap	<code>\$RelationMap[i].contextElement</code>
cutElementNames	Boolean	RelationMap	<code>\$RelationMap[i].cutElementNames</code>
depth	int	RelationMap	<code>\$RelationMap[i].depth</code>
elementType	String	RelationMap	<code>\$RelationMap[i].elementType</code>
expandedElements	String	RelationMap	<code>\$RelationMap[i].expandedElements</code>
filterAreaExpanded	Boolean	RelationMap	<code>\$RelationMap[i].filterAreaExpanded</code>
groupingCriterion	String	RelationMap	<code>\$RelationMap[i].groupingCriterion</code>
includeSubtypes	Boolean	RelationMap	<code>\$RelationMap[i].includeSubtypes</code>
isInitialized	Boolean	RelationMap	<code>\$RelationMap[i].isInitialized</code>
layout	relationMapLayoutEnumeration	RelationMap	<code>\$RelationMap[i].layout</code>
layoutData	String	RelationMap	<code>\$RelationMap[i].layoutData</code>
makeSelectedAsContext	Boolean	RelationMap	<code>\$RelationMap[i].makeSelectedAsContext</code>
relationCriterion	StructuredExpression	RelationMap	<code>\$RelationMap[i].relationCriterion</code>
removedElements	String	RelationMap	<code>\$RelationMap[i].removedElements</code>
scope	Element	RelationMap	<code>\$RelationMap[i].scope</code>
ShowAppliedStereotypes	Boolean	RelationMap	<code>\$RelationMap[i].ShowAppliedStereotypes</code>
showElementNumbers	Boolean	RelationMap	<code>\$RelationMap[i].showElementNumbers</code>
showFullTypes	Boolean	RelationMap	<code>\$RelationMap[i].showFullTypes</code>
showGrouping	Boolean	RelationMap	<code>\$RelationMap[i].showGrouping</code>
showLegend	Boolean	RelationMap	<code>\$RelationMap[i].showLegend</code>
showParameters	Boolean	RelationMap	<code>\$RelationMap[i].showParameters</code>
showRelationStyles	Boolean	RelationMap	<code>\$RelationMap[i].showRelationStyles</code>
showSingleNodePerElement	Boolean	RelationMap	<code>\$RelationMap[i].showSingleNodePerElement</code>
suppressedElements	String	RelationMap	<code>\$RelationMap[i].suppressedElements</code>

725. relationMapLayoutEnumeration

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Radial	Enumeration Literal	relationMapLayoutEnumeration	<code>\$relationMapLayoutEnumeration[i].Radial</code>
Tree	Enumeration Literal	relationMapLayoutEnumeration	<code>\$relationMapLayoutEnumeration[i].Tree</code>

726. RelationOption

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
All	Enumeration Literal	RelationOption	<code>\$RelationOption[i].All</code>
Columns without relations	Enumeration Literal	RelationOption	<code>\$RelationOption[i].Columns without relations</code>
Rows without relations	Enumeration Literal	RelationOption	<code>\$RelationOption[i].Rows without relations</code>
With relations	Enumeration Literal	RelationOption	<code>\$RelationOption[i].With relations</code>
Without relations	Enumeration Literal	RelationOption	<code>\$RelationOption[i].Without relations</code>

727. Releasability

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
distribution unlimited	Enumeration Literal	Releasability	<code>\$Releasability[i].distribution unlimited</code>
DoD access report	Enumeration Literal	Releasability	<code>\$Releasability[i].DoD access report</code>
DoD and DoD contractors only	Enumeration Literal	Releasability	<code>\$Releasability[i].DoD and DoD contractors only</code>
not known	Enumeration Literal	Releasability	<code>\$Releasability[i].not known</code>
not specified	Enumeration Literal	Releasability	<code>\$Releasability[i].not specified</code>
originator controlled	Enumeration Literal	Releasability	<code>\$Releasability[i].originator controlled</code>
report control special dissemination limit	Enumeration Literal	Releasability	<code>\$Releasability[i].report control special dissemination limit</code>
technology data export control	Enumeration Literal	Releasability	<code>\$Releasability[i].technology data export control</code>
US gov agencies and their contractors only	Enumeration Literal	Releasability	<code>\$Releasability[i].US gov agencies and their contractors only</code>
US government access only	Enumeration Literal	Releasability	<code>\$Releasability[i].US government access only</code>

728. ReplaceStereotype

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
disableNewTypeCreation	boolean	ReplaceStereotype	\$ReplaceStereotype[i].disableNewTypeCreation
disableReplaceWhereSavedAsElementValue	boolean	ReplaceStereotype	\$ReplaceStereotype[i].disableReplaceWhereSavedAsElementValue
explicitNewMetaclass	Element	ReplaceStereotype	\$ReplaceStereotype[i].explicitNewMetaclass
isStrict	boolean	ReplaceStereotype	\$ReplaceStereotype[i].isStrict

729. ReplaceTaggedValue

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
addStereotypelfNotAdded	boolean	ReplaceTaggedValue	\$ReplaceTaggedValue[i].addStereotypeIfNotAdded
addToDolfTagDoesNotExist	boolean	ReplaceTaggedValue	\$ReplaceTaggedValue[i].addToDoIfTagDoesNotExist
caseSensitiveEnumerationLiteral	boolean	ReplaceTaggedValue	\$ReplaceTaggedValue[i].caseSensitiveEnumerationLiteral

730. ReplaceType

731. ReportCategory

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
DoDAF	Enumeration Literal	ReportCategory	\$ReportCategory[i].DoDAF
MODAF	Enumeration Literal	ReportCategory	\$ReportCategory[i].MODAF
NAF	Enumeration Literal	ReportCategory	\$ReportCategory[i].NAF
NAF_4_0	Enumeration Literal	ReportCategory	\$ReportCategory[i].NAF_4_0

732. ReportData

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
autoImageSize	AutoImageSize	ReportData	\$ReportData[i].autoImageSize

data	Element	ReportData	<code>\$ReportData[i].data</code>
emptyText	String	ReportData	<code>\$ReportData[i].emptyText</code>
generateRecursively	boolean	ReportData	<code>\$ReportData[i].generateRecursively</code>
imageFormat	ImageFormat	ReportData	<code>\$ReportData[i].imageFormat</code>
template	String	ReportData	<code>\$ReportData[i].template</code>

733. ReportDataMappingRule

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
architectureFrameworkKind	ArchitectureFrameworkKind	ReportDataMappingRule	<code>\$ReportDataMappingRule[i].architectureFrameworkKind</code>
categoryID	String	ReportDataMappingRule	<code>\$ReportDataMappingRule[i].categoryID</code>
packageStereotype	Stereotype	ReportDataMappingRule	<code>\$ReportDataMappingRule[i].packageStereotype</code>
reportDataStereotype	Stereotype	ReportDataMappingRule	<code>\$ReportDataMappingRule[i].reportDataStereotype</code>
resourceID	String	ReportDataMappingRule	<code>\$ReportDataMappingRule[i].resourceID</code>

734. ReportTemplate

Base Classifier

- [AttachedFile](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
attachedAt	date	AttachedFile	<code>\$ReportTemplate[i].attachedAt</code>
author	String	AttachedFile	<code>\$ReportTemplate[i].author</code>
file	String	AttachedFile	<code>\$ReportTemplate[i].file</code>
modifiedAt	date	AttachedFile	<code>\$ReportTemplate[i].modifiedAt</code>
size	String	AttachedFile	<code>\$ReportTemplate[i].size</code>
templateXML	String	ReportTemplate	<code>\$ReportTemplate[i].templateXML</code>

735. Representation Kind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Separate Paragraph	Enumeration Literal	Representation Kind	<code>\$RepresentationKind[i].Separate Paragraph</code>

Single Paragraph	Enumeration Literal	Representation Kind	\$RepresentationKind[i].Single Paragraph
------------------	---------------------	---------------------	--

736. Request

A Request represents a feature of a Participant that is the consumption of a service by one participant provided by others using well-defined terms, conditions and interfaces. A Request designates ports that define the connection point through which a Participant meets its needs through the consumption of services provided by others.

Base Classifier

- [Port](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
connectorRequired	Boolean	Port	\$Request[i].connectorRequired

737. Request

UPDM: From SOAML A Request represents a feature of a Participant that is the consumption of a service by one participant provided by others using well-defined terms, conditions and interfaces. A Request designates ports that define the connection point through which a Participant meets its needs through the consumption of services provided by others.

MODAF: Similar to requires, Asserts that a Resource requires a Service to be provided in order to function correctly.

DoDAF: Similar to ServicePort, A part of a Performer that specifies a distinct interaction point through which the Performer interacts with other Performers. This isolates dependencies between performers to particular interaction points rather than to the performer as a whole.

Base Classifier

- [Request](#)
- [ServicePort](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$Request[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$Request[i].conformsTo
connectorRequired	Boolean	Port	\$Request[i].connectorRequired
endBoundaryType	ISO8601DateTime	UPDMElement	\$Request[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$Request[i].propertySet
providedByResource	ServiceLevelValueSet	ServicePort	\$Request[i].providedByResource
ServicePort.actualPropertySets		ServicePort	\$Request[i].ServicePort.actualPropertySets
ServicePort.type		ServicePort	\$Request[i].ServicePort.type
startBoundaryType	ISO8601DateTime	UPDMElement	\$Request[i].startBoundaryType
URI	String	UPDMElement	\$Request[i].URI

738. Requirement

A requirement specifies a capability or condition that must (or should) be satisfied. A requirement may specify a function that a system must perform or a performance condition that a system must satisfy. Requirements are used to establish a contract between the customer (or other stakeholder) and those responsible for designing and implementing the system.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	\$Requirement[i].Derived
DerivedFrom	Requirement	Requirement	\$Requirement[i].DerivedFrom
Id	String	Requirement	\$Requirement[i].Id
Master	Requirement	Requirement	\$Requirement[i].Master
RefinedBy	NamedElement	Requirement	\$Requirement[i].RefinedBy
SatisfiedBy	NamedElement	Requirement	\$Requirement[i].SatisfiedBy
Text	String	Requirement	\$Requirement[i].Text
TracedTo	NamedElement	Requirement	\$Requirement[i].TracedTo
VerifiedBy	NamedElement	Requirement	\$Requirement[i].VerifiedBy

739. RequirementRelated

This stereotype is used to add properties to those elements that are related to requirements via the various dependencies.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Refines	Requirement	RequirementRelated	\$RequirementRelated[i].Refines
Satisfies	Requirement	RequirementRelated	\$RequirementRelated[i].Satisfies
TracedFrom	Requirement	RequirementRelated	\$RequirementRelated[i].TracedFrom
Verifies	Requirement	RequirementRelated	\$RequirementRelated[i].Verifies

740. RequirementTable

A **Requirements Table** allows you to organize your requirements in a tabular form. As requirements are text-based, this table provides a convenient way for filling-in requirements' information using a spreadsheet-like tabular format, instead of limited-size boxes in a diagram. Each row in the table represents a requirement. The table columns represents the properties of each requirement in the table. With this table, you can:

- Create new requirements directly in the table, or import the existing ones from your model to the table.
- Directly edit the properties of the requirements in the table.
- Directly generate requirement reports, renumber requirements' IDs, or export the table into CSV or HTML format.

There are 3 methods to add requirement(s) in the table:

1. Create a new requirement and add to the table.

- Click the "**Add New**" button on the table toolbar, and then select a requirement types you would like to create from the drop-down menu. The owner of the newly-created requirement will be similar to the owner of the table.
- To select a different owner, hold Shift and then select a requirement type from the drop-down menu.
- Shortcut: **Insert** (**Ctrl+I** on MAC)

2. Create a new nested requirement and add to the table.

- Click the "**Add Nested**" button on the table toolbar while an existing requirement is highlighted in the table, and then select a requirement types you would like to create from the drop-down menu. The owner of the newly-created requirement will be the highlighted requirement.
- Alternatively, just right-click the requirement row in the table, and then select **Add Nested** option in the displayed shortcut menu.
- Shortcut: **Alt+Insert** (**Alt+I** on MAC).

3. Add existing requirement(s) to the table.

- Click "**Add Existing**" button. In the dialog, select requirement(s) already existed in your model to display it(them) in the table.
- Or, directly drag existing requirement(s) from a browser to the table.
- Shortcut: **Ctrl+Insert** (**Ctrl+E** on MAC) Additional commands are available when right-click on a cell in the table.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
columnsWidth	Integer	RequirementTable	\$RequirementTable[i].columnsWidth
customColumnElements	Element	RequirementTable	\$RequirementTable[i].customColumnElements
showColumns	Boolean	RequirementTable	\$RequirementTable[i].showColumns
showCustomColumns	Boolean	RequirementTable	\$RequirementTable[i].showCustomColumns

741. RequiresCompetence

MODAF:: Asserts that an Role requires a Competence (MODAF::CompetenceForRole).

DoDAF: An overlap between a Personnel Type and the Skills it entails (DoDAF:: SkillPartOfPersonType).

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$RequiresCompetence[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$RequiresCompetence[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$RequiresCompetence[i].endBoundaryType</code>
measurementSet	ActualPropertySet	RequiresCompetence	<code>\$RequiresCompetence[i].measurementSet</code>
propertySet	PropertySet	UPDMElement	<code>\$RequiresCompetence[i].propertySet</code>
RequiresCompetence.client		RequiresCompetence	<code>\$RequiresCompetence[i].RequiresCompetence.client</code>
RequiresCompetence.supplier		RequiresCompetence	<code>\$RequiresCompetence[i].RequiresCompetence.supplier</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$RequiresCompetence[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$RequiresCompetence[i].URI</code>

742. Resource

UPDM: Abstract element placeholder to indicate that resources can be exchanged in Operational and Systems views.

MODAF: NA.

DoDAF: Data, Information, Performers, Materiel, or Personnel Types that are produced or consumed.

Base Classifier

- [LocationHolder](#)
- [PropertySet](#)
- [SubjectOfResourceConstraint](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Resource[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Resource[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Resource[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Resource[i].endBoundaryType</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$Resource[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$Resource[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$Resource[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Resource[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Resource[i].URI</code>

743. Resource Impact Analysis Map

*The **Resource Impact Analysis Map** depicts the model elements influenced by the changes of the Resource. The predefined map includes:*

- *Performs Function*
- *Exhibits Capability*
- *Context (Resource used as Context)*
- *Implements*
- *Inputs and Outputs (Resources connected using Resource Interactions)*
- *Resource Interactions*

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:


-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:

Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .


Expand / suppress branches - click on smart manipulator after the Node .

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

744. Resource Role Impact Analysis Map

The **Resource Role Impact Analysis Map** depicts the model elements influenced by the changes of the Resource Role. The predefined map includes:

- *Context (Resources used as the context of the Resource Role)*
- *Inputs and Outputs (Resource Roles connected using Resource Interactions)*

- *Performs In Context (Functions performed by the Resource Role)*
- *Performs (by Type) (Functions performed by the Type of the Resource Role)*
- *Implements (by Type) (Nodes implemented by the Type of the Resource Role)*
- *Exhibits Capability (by Type) (Capabilities exhibited by the Type of the Resource Role)*

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:

Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node  / .

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

745. ResourceAction

Base Classifier

- InvisibleStereotype

746. ResourceArtifact

UPDM: A combination of physical element, energy, and data that are combined used to accomplish a task or function.

MODAF: A type of man-made object. Examples are "car", "radio", "fuel", etc. (MODAF:: Artefact).

Base Classifier

- [PhysicalResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$ResourceArtifact[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceArtifact[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$ResourceArtifact[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$ResourceArtifact[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$ResourceArtifact[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceArtifact[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$ResourceArtifact[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$ResourceArtifact[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceArtifact[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$ResourceArtifact[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$ResourceArtifact[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$ResourceArtifact[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$ResourceArtifact[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceArtifact[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceArtifact[i].URI</code>

747. ResourceArtifactConceptRole

Base Classifier

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceArtifactConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$ResourceArtifactConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$ResourceArtifactConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceArtifactConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceArtifactConceptRole[i].propertySet</code>

startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceArtifactConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceArtifactConceptRole[i].URI</code>

748. ResourceAssociation

Relationship summarizing Resource Interactions between connected Resources. It is not a part of UPDM 2.0 specification.

749. ResourceConnector

UPDM: A physical connection between two resources that implements protocols through which the source resource can transmit items to the destination resource.

MODAF: Asserts that a connection exists between two ports belonging to parts in a system composite structure model (MODAF::SystemPortConnector).

DoDAF: NA

Base Classifier

- [ProtocollImplementation](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceConnector[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ResourceConnector[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceConnector[i].endBoundaryType</code>
implements	Protocol	ProtocollImplementation	<code>\$ResourceConnector[i].implements</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceConnector[i].propertySet</code>
realizedExchange	ResourceInteraction	ResourceConnector	<code>\$ResourceConnector[i].realizedExchange</code>
realizedInterface	ResourceInterface	ResourceConnector	<code>\$ResourceConnector[i].realizedInterface</code>
ResourceConnector.end		ResourceConnector	<code>\$ResourceConnector[i].ResourceConnector.end</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceConnector[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceConnector[i].URI</code>

750. ResourceConstraint

MODAF: A rule governing the structural or functional aspects of an implementation - this may also include constraints on OrganisationalResources that are part of an implementation.

DoDAF: The range of permissible states for an object (DoDAF::Constraint).

Base Classifier

- [Rule](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceConstraint[i].actualPropertySet</code>

conformsTo	Standard	UPDMElement	\$ResourceConstraint[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceConstraint[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ResourceConstraint[i].propertySet
ResourceConstraint.constrainedElement		ResourceConstraint	\$ResourceConstraint[i].ResourceConstraint.constrainedElement
ruleKind	RuleKind	Rule	\$ResourceConstraint[i].ruleKind
startBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceConstraint[i].startBoundaryType
URI	String	UPDMElement	\$ResourceConstraint[i].URI

751. ResourceEventTrace

UPDM: A UPDM artifact that extends a UML Interaction.

Base Classifier

- UPDMElement

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ResourceEventTrace[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ResourceEventTrace[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceEventTrace[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ResourceEventTrace[i].propertySet
ResourceEventTrace.message		ResourceEventTrace	\$ResourceEventTrace[i].ResourceEventTrace.message
ResourceEventTrace.owner		ResourceEventTrace	\$ResourceEventTrace[i].ResourceEventTrace.owner
startBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceEventTrace[i].startBoundaryType
URI	String	UPDMElement	\$ResourceEventTrace[i].URI

752. ResourceInteraction

UPDM: ResourceInteraction represents data that is exchanged between the resources

MODAF: An assertion that two FunctionalResources interact. Examples : data exchange between systems, conversations between people, people using systems.

DoDAF: NA

Base Classifier

- Exchange
- SubjectOfResourceConstraint

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ResourceInteraction[i].actualPropertySet

conformsTo	Standard	UPDMElement	\$ResourceInteraction[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceInteraction[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ResourceInteraction[i].propertySet
ResourceInteraction.conveyedElement		ResourceInteraction	\$ResourceInteraction[i].ResourceInteraction.conveyedElement
ResourceInteraction.informationSource		ResourceInteraction	\$ResourceInteraction[i].ResourceInteraction.informationSource
ResourceInteraction.informationTarget		ResourceInteraction	\$ResourceInteraction[i].ResourceInteraction.informationTarget
ResourceInteraction.realization		ResourceInteraction	\$ResourceInteraction[i].ResourceInteraction.realization
ResourceInteraction.realizingActivityEdge		ResourceInteraction	\$ResourceInteraction[i].ResourceInteraction.realizingActivityEdge
ResourceInteraction.realizingConnector		ResourceInteraction	\$ResourceInteraction[i].ResourceInteraction.realizingConnector
startBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceInteraction[i].startBoundaryType
URI	String	UPDMElement	\$ResourceInteraction[i].URI

753. ResourceInteractionItem

UPDM Abstract: Represents the item(s) exchanged between the resources through a ResourceInteraction.

MODAF: Formalised representation of data which is managed by or exchanged between systems (MODAF::DataElement).

DoDAF: Representation of information in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means (DoDAF::Data).

Base Classifier

- Resource

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ResourceInteractionItem[i].actualPropertySet
affectedFunctions	Function	ResourceInteractionItem	\$ResourceInteractionItem[i].affectedFunctions
appliesTo	UPDMElement	PropertySet	\$ResourceInteractionItem[i].appliesTo
conformsTo	Standard	UPDMElement	\$ResourceInteractionItem[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceInteractionItem[i].endBoundaryType
physicalLocation	ActualLocation	LocationHolder	\$ResourceInteractionItem[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$ResourceInteractionItem[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$ResourceInteractionItem[i].requiredEnvironment

startBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceInteractionItem[i].startBoundaryType
URI	String	UPDMElement	\$ResourceInteractionItem[i].URI

754. ResourceInterface

UPDM: ResourceInterface is a contractual agreement between two resources that implement protocols through which the source resource to the destination resource.

MODAF: NA

DoDAF: An overlap between Performers for the purpose of producing a Resource that is consumed by the other (DoDAF:: Interface).

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ResourceInterface[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ResourceInterface[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceInterface[i].endBoundaryType
identifier	String	ResourceInterface	\$ResourceInterface[i].identifier
propertySet	PropertySet	UPDMElement	\$ResourceInterface[i].propertySet
realizedExchange	ResourceInteraction	ResourceInterface	\$ResourceInterface[i].realizedExchange
realizingConnector	ResourceConnector	ResourceInterface	\$ResourceInterface[i].realizingConnector
ResourceInterface.end		ResourceInterface	\$ResourceInterface[i].ResourceInterface.end
startBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceInterface[i].startBoundaryType
URI	String	UPDMElement	\$ResourceInterface[i].URI

755. ResourceMessage

UPDM: Message for use in a Resource Event-Trace, implements a ResourceInteraction.

MODAF: A specification of the interactions between aspects of a Resources architecture (MODAF::ResourceInteractionSpecification).

DoDAF: An overlap of an Activity with a Resource, in particular a consuming or producing Activity that expresses an input, output, consumption, or production Activity of the Resource (DoDAF:: activityResourceOverlap).

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ResourceMessage[i].actualPropertySet
carries	ResourceInteraction	ResourceMessage	\$ResourceMessage[i].carries
conformsTo	Standard	UPDMElement	\$ResourceMessage[i].conformsTo

endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceMessage[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceMessage[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceMessage[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceMessage[i].URI</code>

756. ResourceOperation

UPDM: A partial or full realization of Function.

MODAF: NA

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceOperation[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ResourceOperation[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceOperation[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceOperation[i].propertySet</code>
realizes	Function	ResourceOperation	<code>\$ResourceOperation[i].realizes</code>
ResourceOperation.ownedParameter		ResourceOperation	<code>\$ResourceOperation[i].ResourceOperation.ownedParameter</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceOperation[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceOperation[i].URI</code>

757. ResourceParameter

UPDM: Represents inputs and outputs of Function. It is typed by ResourceInteractionItem.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceParameter[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ResourceParameter[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceParameter[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceParameter[i].propertySet</code>
ResourceParameter.type		ResourceParameter	<code>\$ResourceParameter[i].ResourceParameter.type</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceParameter[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceParameter[i].URI</code>

758. ResourcePort

UPDM: Port is an interaction point for a resource through which it can interact with the outside environment.

MODAF: An interface (logical or physical) provided by a System. A SystemPort may implement a PortType though there is no requirement for SystemPorts to be typed (MODAF:: SystemPort).

DoDAF: An interface (logical or physical) provided by a System (DoDAF::Port).

Base Classifier

- [ProtocollImplementation](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourcePort[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ResourcePort[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourcePort[i].endBoundaryType</code>
implements	Protocol	ProtocollImplementation	<code>\$ResourcePort[i].implements</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourcePort[i].propertySet</code>
ResourcePort.type		ResourcePort	<code>\$ResourcePort[i].ResourcePort.type</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourcePort[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourcePort[i].URI</code>

759. ResourceRole

UPDM: abstract element.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceRole[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ResourceRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceRole[i].endBoundaryType</code>
performsInContext	Function	ResourceRole	<code>\$ResourceRole[i].performsInContext</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceRole[i].propertySet</code>
ResourceRole.type		ResourceRole	<code>\$ResourceRole[i].ResourceRole.type</code>
ResourceRole.class		ResourceRole	<code>\$ResourceRole[i].ResourceRole.class</code>
roleKind	RoleKind	ResourceRole	<code>\$ResourceRole[i].roleKind</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceRole[i].URI</code>

760. ResourceRoleMapping

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Context	Stereotype	ResourceRoleMapping	<code>\$ResourceRoleMapping[i].Context</code>
expectContext	Stereotype	ResourceRoleMapping	<code>\$ResourceRoleMapping[i].expectContext</code>
expectTypes	Stereotype	ResourceRoleMapping	<code>\$ResourceRoleMapping[i].expectTypes</code>
RoleKind	RoleKind	ResourceRoleMapping	<code>\$ResourceRoleMapping[i].RoleKind</code>
Types	Stereotype	ResourceRoleMapping	<code>\$ResourceRoleMapping[i].Types</code>

761. ResourceRoleTable

Base Classifier

- [InvisibleStereotype](#)

762. ResourceState

UPDM: State identified in the context of an *ResourceStateDescription*.

MODAF:N/A

DoDAF:N/A

Base Classifier

- [DesiredState](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ResourceState[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ResourceState[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceState[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ResourceState[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ResourceState[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ResourceState[i].URI</code>

763. ResourceStateMachine

UPDM Artifact that extends a UML *StateMachine* allied to Resources.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$ResourceStateMachine[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$ResourceStateMachine[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceStateMachine[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$ResourceStateMachine[i].propertySet
ResourceStateMachine.owner		ResourceStateMachine	\$ResourceStateMachine[i].ResourceStateMachine.owner
startBoundaryType	ISO8601DateTime	UPDMElement	\$ResourceStateMachine[i].startBoundaryType
URI	String	UPDMElement	\$ResourceStateMachine[i].URI

764. Responsibility

A contract or an obligation of an element in its relationship to other elements.

765. Responsibility

UPDM: Asserts that a Post or Organization has specific responsibilities.

MODAF: NA

DoDAF: NA

Base Classifier

- [CompetenceRequirer](#)
- [OrganizationalResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	\$Responsibility[i].actsUpon
actualPropertySet	ActualPropertySet	UPDMElement	\$Responsibility[i].actualPropertySet
affectedFunctions	Function	ResourceInteractionItem	\$Responsibility[i].affectedFunctions
appliesTo	UPDMElement	PropertySet	\$Responsibility[i].appliesTo
conformsTo	Standard	UPDMElement	\$Responsibility[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Responsibility[i].endBoundaryType
milestone	ActualProjectMilestone	SystemResource	\$Responsibility[i].milestone
physicalLocation	ActualLocation	LocationHolder	\$Responsibility[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$Responsibility[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$Responsibility[i].requiredEnvironment
Resource.ownedOperation		SystemResource	\$Responsibility[i].Resource.ownedOperation
Resource.ownedPort		SystemResource	\$Responsibility[i].Resource.ownedPort
Resource.performs		SystemResource	\$Responsibility[i].Resource.performs

startBoundaryType	ISO8601DateTime	UPDMElement	\$Responsibility[i].startBoundaryType
URI	String	UPDMElement	\$Responsibility[i].URI

766. RiskKind

- 1) High indicates an unacceptable level of risk,
- 2) Medium indicates an acceptable level of risk, and
- 3) Low indicates a minimal level of risk or no risk

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
High	Enumeration Literal	RiskKind	\$RiskKind[i].High
Low	Enumeration Literal	RiskKind	\$RiskKind[i].Low
Medium	Enumeration Literal	RiskKind	\$RiskKind[i].Medium

767. RoleKind

Enumeration of the roles that a ResourceRole may play in the context of a CapabilityConfiguration or System, used to support the RoleKind tag of a ResourceRole.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Component	Enumeration Literal	RoleKind	\$RoleKind[i].Component
Equipment	Enumeration Literal	RoleKind	\$RoleKind[i].Equipment
Hosted Software	Enumeration Literal	RoleKind	\$RoleKind[i].Hosted Software
Human Resource	Enumeration Literal	RoleKind	\$RoleKind[i].Human Resource
Other	Enumeration Literal	RoleKind	\$RoleKind[i].Other
Part	Enumeration Literal	RoleKind	\$RoleKind[i].Part
Platform	Enumeration Literal	RoleKind	\$RoleKind[i].Platform
Post Role	Enumeration Literal	RoleKind	\$RoleKind[i].Post Role
Responsibility Role	Enumeration Literal	RoleKind	\$RoleKind[i].Responsibility Role
Service Access Role	Enumeration Literal	RoleKind	\$RoleKind[i].Service Access Role
Sub Organization	Enumeration Literal	RoleKind	\$RoleKind[i].Sub Organization
Sub System Part	Enumeration Literal	RoleKind	\$RoleKind[i].Sub System Part
System	Enumeration Literal	RoleKind	\$RoleKind[i].System
Used Configuration	Enumeration Literal	RoleKind	\$RoleKind[i].Used Configuration

768. RoleType

MODAF: An aspect of a person or organization that enables them to fulfill a particular function.

Base Classifier

- [Responsibility](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$RoleType[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$RoleType[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$RoleType[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$RoleType[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$RoleType[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$RoleType[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$RoleType[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$RoleType[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$RoleType[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$RoleType[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$RoleType[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$RoleType[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$RoleType[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$RoleType[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$RoleType[i].URI</code>

769. Rule

MODAF: An abstract Class that is extended by [OperationalConstraint](#) (A rule governing an operational behaviour or property.) and [ResourceConstraint](#) (A rule governing the structural or functional aspects of an implementation - this may also include constraints on [OrganisationalResources](#) that are part of an implementation).

DoDAF: Rule: A principle or condition that governs behavior; a prescribed guide for conduct or action. Subtype: [Constraint](#): The range of permissible states for an object.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Rule[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Rule[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Rule[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Rule[i].propertySet</code>
ruleKind	RuleKind	Rule	<code>\$Rule[i].ruleKind</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Rule[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Rule[i].URI</code>

770. RuleKind

Enumeration of possible kinds for constraints.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ActionAssertion	Enumeration Literal	RuleKind	<code>\$RuleKind[i].ActionAssertion</code>
Agreement	Enumeration Literal	RuleKind	<code>\$RuleKind[i].Agreement</code>
Constraint	Enumeration Literal	RuleKind	<code>\$RuleKind[i].Constraint</code>
Derivation	Enumeration Literal	RuleKind	<code>\$RuleKind[i].Derivation</code>
Guidance	Enumeration Literal	RuleKind	<code>\$RuleKind[i].Guidance</code>
SecurityPolicy	Enumeration Literal	RuleKind	<code>\$RuleKind[i].SecurityPolicy</code>
StructuralAssertion	Enumeration Literal	RuleKind	<code>\$RuleKind[i].StructuralAssertion</code>

771. S1

Base Classifier

- [InvisibleStereotype](#)

772. S1 Package

Base Classifier

- [InvisibleStereotype](#)

773. S3

Base Classifier

- [InvisibleStereotype](#)

774. S3 Package

Base Classifier

- [InvisibleStereotype](#)

775. S3 Report

Base Classifier

- [InvisibleStereotype](#)

776. S4

Base Classifier

- [InvisibleStereotype](#)

777. S4 Package

Base Classifier

- [InvisibleStereotype](#)

778. S4 Services to Operational Activities Mapping

Services to Operational Activities Mapping (S4) purpose is to provide traceability by illustrating which services support which operational activities..

The Rows of this matrix are Service Interfaces and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Service Interfaces);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Operational Activity maps to Capability ("Maps To Capability" relationship) that is exposed by Service Interface ("Expose" relationship).

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

hideColumns	String	S4 Services to Operational Activities Mapping	\$S4ServicestoOperationalActivitiesMapping[i].hideColumns
-------------	--------	---	---

779. S5

Base Classifier

- [InvisibleStereotype](#)

780. S5 Package

Base Classifier

- [InvisibleStereotype](#)

781. S6

Base Classifier

- [InvisibleStereotype](#)

782. S6 Package

Base Classifier

- [InvisibleStereotype](#)

783. S7

Base Classifier

- [InvisibleStereotype](#)

784. S7 Package

Base Classifier

- [InvisibleStereotype](#)

785. S8

Service Policy (S8) table specifies constraints that apply to implementations of services. There are two ways to fill this table.

- 1. Add new Service Policy. Click **Add New** button and select constrained Service Interface. Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. Add Existing Service Policy. Click **Add Existing** button and select Service Policies.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Service Policy) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	S8	\$S8[i].hideColumns

786. S8 Package

Base Classifier

- [InvisibleStereotype](#)

787. S8 Report

Base Classifier

- [InvisibleStereotype](#)

788. SameAs

MODAF: Asserts that two elements refer to the same real-world thing.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SameAs[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$SameAs[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SameAs[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$SameAs[i].propertySet</code>
SameAs.client		SameAs	<code>\$SameAs[i].SameAs.client</code>
SameAs.supplier		SameAs	<code>\$SameAs[i].SameAs.supplier</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SameAs[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$SameAs[i].URI</code>

789. Satisfy

A Satisfy relationship is a dependency between a requirement and a model element that fulfills the requirement. As with other dependencies, the arrow direction points from the satisfying (client) model element to the (supplier) requirement that is satisfied.

Base Classifier

- [Trace](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getSatisfies	Requirement	Satisfy	<code>\$Satisfy[i].getSatisfies</code>
getTracedFrom	Requirement	Trace	<code>\$Satisfy[i].getTracedFrom</code>
sourceContext	Classifier	DirectedRelationshipPropertyPath	<code>\$Satisfy[i].sourceContext</code>
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$Satisfy[i].sourcePropertyPath</code>
targetContext	Classifier	DirectedRelationshipPropertyPath	<code>\$Satisfy[i].targetContext</code>
targetPropertyPath	Property	DirectedRelationshipPropertyPath	<code>\$Satisfy[i].targetPropertyPath</code>

790. Script

A script file that can be interpreted by a computer system.
Subclass of «file».

Base Classifier

- [File](#)

791. SDDSubsystem

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
subsystemInteractions	String	SDDSubsystem	<code>\$SDDSubsystem[i].subsystemInteractions</code>
subsystemResources	String	SDDSubsystem	<code>\$SDDSubsystem[i].subsystemResources</code>

792. SecurityAttributesGroup

MODAF:NA

DoDAF: The group of Information Security Marking attributes in which the use of attributes 'classification' and 'ownerProducer' is required. This group is to be contrasted with group 'SecurityAttributesOptionGroup' in which use of those attributes is optional.

Base Classifier

- [PropertySet](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SecurityAttributesGroup[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$SecurityAttributesGroup[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$SecurityAttributesGroup[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SecurityAttributesGroup[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$SecurityAttributesGroup[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SecurityAttributesGroup[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$SecurityAttributesGroup[i].URI</code>

793. SecurityClassification

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
confidential	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].confidential</code>
confidential restricted	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].confidential restricted</code>
confidential/no foreign	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].confidential/no foreign</code>
for official use only	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].for official use only</code>
NATO confidential	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO confidential</code>

NATO confidential atomal	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO confidential atomal</code>
NATO restricted	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO restricted</code>
NATO secret	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO secret</code>
NATO secret atomal	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO secret atomal</code>
NATO TOP secret	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO TOP secret</code>
NATO TOP secret atomal	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO TOP secret atomal</code>
NATO unclassified	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].NATO unclassified</code>
no classification	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].no classification</code>
secret	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].secret</code>
secret restricted	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].secret restricted</code>
secret/no foreign	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].secret/no foreign</code>
TOP secret	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].TOP secret</code>
unclassified	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].unclassified</code>
unclassified sensitive	Enumeration Literal	SecurityClassification	<code>\$SecurityClassification[i].unclassified sensitive</code>

794. SecurityDomain

MODAF:NA

DoDAF: A NodeType whose members (other Nodes, KnownResources) all share a common security policy.

Base Classifier

- [Node](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$SecurityDomain[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SecurityDomain[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$SecurityDomain[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$SecurityDomain[i].conformsTo</code>
connectedNodes	Node	Node	<code>\$SecurityDomain[i].connectedNodes</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SecurityDomain[i].endBoundaryType</code>
Node.ownedPort		Node	<code>\$SecurityDomain[i].Node.ownedPort</code>
Node.performs		Node	<code>\$SecurityDomain[i].Node.performs</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$SecurityDomain[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$SecurityDomain[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$SecurityDomain[i].requiredEnvironment</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SecurityDomain[i].startBoundaryType</code>
SubjectOfOperationalStateMachin		SubjectOfOperationalStateMachin	<code>\$SecurityDomain[i].SubjectOfOperationalState</code>

e.ownedBehavior		e	Machine.ownedBehavior
URI	String	UPDMElement	\$SecurityDomain[i].URI

795. Semantic

A specialization of “InformationElement” that enables the specification of a complete dataset, which is considered meaningful to a community, organization, system or application; meeting one or more of the information flow requirements specification for a needline. The semantic is defined by the community, needline or application interface.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
containedTransactionals	Transactional	Semantic	\$Semantic[i].containedTransactionals
identifier	Transactional	Semantic	\$Semantic[i].identifier
Semantic.ownedAttribute		Semantic	\$Semantic[i].Semantic.ownedAttribute

796. SemanticAttribute

Specialization of Entity Attribute that enables the relationship between logical/Interim-Processing and Operational/Business naming conventions.

797. Send

A usage dependency whose source is an operation and whose target is a signal, specifying that the source sends the target signal.

798. Sensor

A Sensor is a special external system that forwards information from the environment to the system under development. For example a Temperature sensor.

Base Classifier

- [External system](#)

799. SequenceType

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Character	Enumeration Literal	SequenceType	\$SequenceType[i].Character
Expression	Enumeration Literal	SequenceType	\$SequenceType[i].Expression
Numeric	Enumeration Literal	SequenceType	\$SequenceType[i].Numeric
OwnerNumber	Enumeration Literal	SequenceType	\$SequenceType[i].OwnerNumber

Separator	Enumeration Literal	SequenceType	<code>\$SequenceType [i] .Separator</code>
-----------	---------------------	------------------------------	--

800. Service

A stateless, functional component (computes a value).

801. Service

MODAF: A type of delivered functionality, specified independently of the resources that provide it.

DoDAF: mechanism to enable access to a set of one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. The mechanism is a Performer. The "capabilities" accessed are Resources -- Information, Data, Materiel, Performers, and Geo-political Extents.

Base Classifier

- [Service](#)
- [ServicePort](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Service [i] .actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Service [i] .conformsTo</code>
connectorRequired	Boolean	Port	<code>\$Service [i] .connectorRequired</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Service [i] .endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Service [i] .propertySet</code>
providedByResource	ServiceLevelValueSet	ServicePort	<code>\$Service [i] .providedByResource</code>
ServicePort.actualPropertySets		ServicePort	<code>\$Service [i] .ServicePort .actualPropertySets</code>
ServicePort.type		ServicePort	<code>\$Service [i] .ServicePort .type</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Service [i] .startBoundaryType</code>
URI	String	UPDMElement	<code>\$Service [i] .URI</code>

802. Service

A Service represents a feature of a Participant that is the offer of a service by one participant to others using well defined terms, conditions and interfaces. A Service designates a Port that defines the connection point through which a Participant offers its capabilities and provides a service to clients.

Base Classifier

- [Port](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
connectorRequired	Boolean	Port	<code>\$Service [i] .connectorRequired</code>

803. Service Oriented Viewpoint

Base Classifier

- [InvisibleStereotype](#)

804. Service Viewpoint

Base Classifier

- [InvisibleStereotype](#)

805. ServiceAccess

Base Classifier

- [SystemResource](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$ServiceAccess[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceAccess[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$ServiceAccess[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$ServiceAccess[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceAccess[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceAccess[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$ServiceAccess[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$ServiceAccess[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceAccess[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$ServiceAccess[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$ServiceAccess[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$ServiceAccess[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$ServiceAccess[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceAccess[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceAccess[i].URI</code>

806. ServiceAction

Base Classifier

- [InvisibleStereotype](#)

807. ServiceAttribute

MODAF: A property of Service.

DoDAF: NA

Base Classifier

- [Property](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceAttribute[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceAttribute[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceAttribute[i].endBoundaryType</code>
maxValue	String	Property	<code>\$ServiceAttribute[i].maxValue</code>
minValue	String	Property	<code>\$ServiceAttribute[i].minValue</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceAttribute[i].propertySet</code>
ServiceAttribute.type		ServiceAttribute	<code>\$ServiceAttribute[i].ServiceAttribute.type</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceAttribute[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceAttribute[i].URI</code>

808. ServiceChannel

A communication path between Services and Requests within an architecture.

809. ServiceContract

A ServiceContract is the formalization of a binding exchange of information, goods, or obligations between parties defining a service.

Base Classifier

- [Collaboration](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isStrict	Boolean	Collaboration	<code>\$ServiceContract[i].isStrict</code>

810. ServiceDescription**Base Classifier**

- ArchitecturalDescription

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceDescription[i].actualPropertySet</code>
approvalAuthority	String	ArchitecturalDescription	<code>\$ServiceDescription[i].approvalAuthority</code>
architect	String	ArchitecturalDescription	<code>\$ServiceDescription[i].architect</code>
ArchitecturalDescription.architectureFramework		ArchitecturalDescription	<code>\$ServiceDescription[i].ArchitecturalDescription.architectureFramework</code>
architectureFramework	ArchitectureFrameworkKind	ArchitecturalDescription	<code>\$ServiceDescription[i].architectureFramework</code>
assumptionAndConstraint	String	ArchitecturalDescription	<code>\$ServiceDescription[i].assumptionAndConstraint</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceDescription[i].conformsTo</code>
creatingOrganization	String	ArchitecturalDescription	<code>\$ServiceDescription[i].creatingOrganization</code>
dateCompleted	String	ArchitecturalDescription	<code>\$ServiceDescription[i].dateCompleted</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceDescription[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceDescription[i].propertySet</code>
purpose	String	ArchitecturalDescription	<code>\$ServiceDescription[i].purpose</code>
recommendations	String	ArchitecturalDescription	<code>\$ServiceDescription[i].recommendations</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceDescription[i].startBoundaryType</code>
summaryOfFindings	String	ArchitecturalDescription	<code>\$ServiceDescription[i].summaryOfFindings</code>
toBe	Boolean	ArchitecturalDescription	<code>\$ServiceDescription[i].toBe</code>
toolsUsed	String	ArchitecturalDescription	<code>\$ServiceDescription[i].toolsUsed</code>
URI	String	UPDMElement	<code>\$ServiceDescription[i].URI</code>
viewpoint	String	ArchitecturalDescription	<code>\$ServiceDescription[i].viewpoint</code>
views	View	ArchitecturalDescription	<code>\$ServiceDescription[i].views</code>

811. ServiceFeature

UPDM:Abstract grouping used to ServiceFunctions to Serviceoperations and ServiceMessageHandlers.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceFeature[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceFeature[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFeature[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceFeature[i].propertySet</code>
ServiceFeature.ownedParameter		ServiceFeature	<code>\$ServiceFeature[i].ServiceFeature.ownedParameter</code>
ServiceFeature.owner		ServiceFeature	<code>\$ServiceFeature[i].ServiceFeature.owner</code>

startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFeature[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceFeature[i].URI</code>

812. ServiceFunction

UPDM: A ServiceFunction describes the abstract behavior of ServiceOperations, regardless of the actual implementation.

MODAF: A type of activity describing the functionality of a service.

DoDAF: Information necessary to interact with the service in such terms as the service inputs, outputs, and associated semantics. The service description also conveys what is accomplished when the service is invoked and the conditions for using the service.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceFunction[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceFunction[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFunction[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceFunction[i].propertySet</code>
ServiceFunction.ownedParameter		ServiceFunction	<code>\$ServiceFunction[i].ServiceFunction.ownedParameter</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFunction[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceFunction[i].URI</code>

813. ServiceFunctionAction

UPDM: A call behavior action that invokes the ServiceFunction that needs to be preformed. --This concept is required for mapping the architecture with UML and does not have a DoDAF or MoDAF equivalent.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceFunctionAction[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceFunctionAction[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFunctionAction[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceFunctionAction[i].propertySet</code>
ServiceFunctionAction.activity		ServiceFunctionAction	<code>\$ServiceFunctionAction[i].ServiceFunctionAction.activity</code>
ServiceFunctionAction.behavior		ServiceFunctionAction	<code>\$ServiceFunctionAction[i].ServiceFunctionAction.behavior</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFunctionAction[i].startBoundaryType</code>

URI	String	UPDMElement	<code>\$ServiceFunctionAction[i].URI</code>
-----	--------	-----------------------------	---

814. ServiceFunctionEdge

UPDM: An extension of <<ActivityEdge>> that is used to model the flow of control/objects through a ServiceFunction.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceFunctionEdge[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceFunctionEdge[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFunctionEdge[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceFunctionEdge[i].propertySet</code>
ServiceFunctionEdge.owner		ServiceFunctionEdge	<code>\$ServiceFunctionEdge[i].ServiceFunctionEdge.owner</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceFunctionEdge[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceFunctionEdge[i].URI</code>

815. ServiceInteraction

UPDM: Interaction for a service interface

MODAF: A model representing how a set of Service classes interacts with one another (MODAF::ServiceInteractionSpecification).

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceInteraction[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceInteraction[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceInteraction[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceInteraction[i].propertySet</code>
ServiceInteraction.message		ServiceInteraction	<code>\$ServiceInteraction[i].ServiceInteraction.message</code>
ServiceInteraction.owner		ServiceInteraction	<code>\$ServiceInteraction[i].ServiceInteraction.owner</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceInteraction[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceInteraction[i].URI</code>

816. ServiceInterface

Provides the definition of a service. Defines the specification of a service interaction as the type of a «Service» or «Request» port.

817. ServiceInterface

UPDM: A contractual agreement between two resources that implement protocols through which the source service interacts to the destination resource. A physical connection between two resources that implements protocols through which the source resource can transmit items to the destination resource.

MODAF: The mechanism by which a Service communicates.

DoDAF: An overlap between Performers for the purpose of producing a Resource that is consumed by the other. (DoDAF::Interface).

SOAML: Defines the interface to a Service Point or Request Point and is the type of a role in a service contract.

Base Classifier

- [PropertySet](#)
- [ServiceInterface](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceInterface[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$ServiceInterface[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceInterface[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceInterface[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceInterface[i].propertySet</code>
serviceInteraction	ServiceInteraction	ServiceInterface	<code>\$ServiceInterface[i].serviceInteraction</code>
ServiceInterface.feature		ServiceInterface	<code>\$ServiceInterface[i].ServiceInterface.feature</code>
ServiceInterface.ownedAttribute		ServiceInterface	<code>\$ServiceInterface[i].ServiceInterface.ownedAttribute</code>
ServiceInterface.ownedRule		ServiceInterface	<code>\$ServiceInterface[i].ServiceInterface.ownedRule</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceInterface[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceInterface[i].URI</code>

818. ServiceLevelValue

MODAF: A ServiceAttributes indicating the level to which a Resource delivers a Service, in a particular environment.

DoDAF: NA

Base Classifier

- [ActualProperty](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceLevelValue[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceLevelValue[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceLevelValue[i].endBoundaryType</code>
endDate	ISO8601DateTime	ActualProperty	<code>\$ServiceLevelValue[i].endDate</code>
intention	ActualPropertySetKind	ActualProperty	<code>\$ServiceLevelValue[i].intention</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceLevelValue[i].propertySet</code>
PropertyValue.definingFeature		ActualProperty	<code>\$ServiceLevelValue[i].PropertyValue.definingFeature</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceLevelValue[i].startBoundaryType</code>
startDate	ISO8601DateTime	ActualProperty	<code>\$ServiceLevelValue[i].startDate</code>
URI	String	UPDMElement	<code>\$ServiceLevelValue[i].URI</code>

819. ServiceLevelValueSet

MODAF: A value specification for a set of ServiceAttributes indicating the level to which a Resource delivers a Service, in a particular environment.

DoDAF: NA

Base Classifier

- [ActualPropertySet](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceLevelValueSet[i].actualPropertySet</code>
ActualPropertySet.classifier		ActualPropertySet	<code>\$ServiceLevelValueSet[i].ActualPropertySet.classifier</code>
ActualPropertySet.slot		ActualPropertySet	<code>\$ServiceLevelValueSet[i].ActualPropertySet.slot</code>
appliesTo	UPDMElement	ActualPropertySet	<code>\$ServiceLevelValueSet[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceLevelValueSet[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceLevelValueSet[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceLevelValueSet[i].propertySet</code>
resourceBoundary	ServicePort	ServiceLevelValueSet	<code>\$ServiceLevelValueSet[i].resourceBoundary</code>
ServiceLevelValueSet.slot		ServiceLevelValueSet	<code>\$ServiceLevelValueSet[i].ServiceLevelValueSet.slot</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceLevelValueSet[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceLevelValueSet[i].URI</code>

820. ServiceMessage

UPDM: Message for use in a Service Interaction Specification, implements a resourceInteraction or any of the subtypes.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceMessage[i].actualPropertySet</code>
carries	Exchange	ServiceMessage	<code>\$ServiceMessage[i].carries</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceMessage[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceMessage[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceMessage[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceMessage[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceMessage[i].URI</code>

821. ServiceMessageHandler

UPDM: An instance of an AsynchronousMessage, applied in the service domain.

Base Classifier

- [ServiceFeature](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceMessageHandler[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceMessageHandler[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceMessageHandler[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceMessageHandler[i].propertySet</code>
ServiceFeature.ownedParameter		ServiceFeature	<code>\$ServiceMessageHandler[i].ServiceFeature.ownedParameter</code>
ServiceFeature.owner		ServiceFeature	<code>\$ServiceMessageHandler[i].ServiceFeature.owner</code>
ServiceMessageHandler.signal		ServiceMessageHandler	<code>\$ServiceMessageHandler[i].ServiceMessageHandler.signal</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceMessageHandler[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceMessageHandler[i].URI</code>

822. ServiceOperation

UPDM: A ServiceOperation provides the access point for invoking the behavior of a provided service. The ServiceOperations are defined on ServiceInterfaces and mirrored on the providing Resource to handle calls forwarded on by the interface.

MODAF: a function or procedure which enables programmatic communication with a Service via a ServiceInterface (MODAF:: ServiceInterfaceOperation).

Base Classifier

- [ServiceFeature](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
abstractBehavior	ServiceFunction	ServiceOperation	<code>\$ServiceOperation[i].abstractBehavior</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceOperation[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceOperation[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceOperation[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceOperation[i].propertySet</code>
ServiceFeature.ownedParameter		ServiceFeature	<code>\$ServiceOperation[i].ServiceFeature.ownedParameter</code>
ServiceFeature.owner		ServiceFeature	<code>\$ServiceOperation[i].ServiceFeature.owner</code>
ServiceOperation.ownedParameter		ServiceOperation	<code>\$ServiceOperation[i].ServiceOperation.ownedParameter</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceOperation[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceOperation[i].URI</code>

823. ServiceParameter

UPDM: Represents inputs and outputs of Service. It is typed by ResourceInteractionItem.

MODAF: A constant or variable passed into or out of a ServiceInterface as part of the execution of a ServiceInterfaceOperation (MODAF:: ServiceInterfaceParameter).

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceParameter[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceParameter[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceParameter[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceParameter[i].propertySet</code>
ServiceParameter.type		ServiceParameter	<code>\$ServiceParameter[i].ServiceParameter.type</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceParameter[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceParameter[i].URI</code>

824. ServicePolicy

UPDM: A constraint governing the consumers and providers of services

MODAF: A constraint governing one or more Services.

DoDAF: Agreement: A consent among parties regarding the terms and conditions of activities that said parties participate in.

Base Classifier

- [Rule](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServicePolicy[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServicePolicy[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServicePolicy[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServicePolicy[i].propertySet</code>
ruleKind	RuleKind	Rule	<code>\$ServicePolicy[i].ruleKind</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServicePolicy[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServicePolicy[i].URI</code>

825. ServicePort

MODAF:ServiceInterface, The mechanism by which a Service communicates.

DoDAF: A part of a Performer that specifies a distinct interaction point through which the Performer interacts with other Performers. This isolates dependencies between performers to particular interaction points rather than to the performer as a whole.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServicePort[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServicePort[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServicePort[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServicePort[i].propertySet</code>
providedByResource	ServiceLevelValueSet	ServicePort	<code>\$ServicePort[i].providedByResource</code>
ServicePort.actualPropertySets		ServicePort	<code>\$ServicePort[i].ServicePort.actualPropertySets</code>
ServicePort.type		ServicePort	<code>\$ServicePort[i].ServicePort.type</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServicePort[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServicePort[i].URI</code>

826. Services Viewpoint

Base Classifier

- [InvisibleStereotype](#)

827. ServicesArchitecture

The high-level view of a Service Oriented Architecture that defines how a set of participants works together, forming a community, for some purpose by providing and using services.

Base Classifier

- [Collaboration](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isStrict	Boolean	Collaboration	<code>\$ServicesArchitecture[i].isStrict</code>

828. ServiceStateMachine

UPDM Artifact that extends a UML StateMachine.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$ServiceStateMachine[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$ServiceStateMachine[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceStateMachine[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$ServiceStateMachine[i].propertySet</code>
ServiceStateMachine.owner		ServiceStateMachine	<code>\$ServiceStateMachine[i].ServiceStateMachine.owner</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$ServiceStateMachine[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$ServiceStateMachine[i].URI</code>

829. setter

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getter/setter for attribute	Element	setter	<code>\$setter[i].getter/setter for attribute</code>

830. SeverityKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
debug	Enumeration Literal	SeverityKind	<code>\$SeverityKind[i].debug</code>
error	Enumeration Literal	SeverityKind	<code>\$SeverityKind[i].error</code>
fatal	Enumeration Literal	SeverityKind	<code>\$SeverityKind[i].fatal</code>
info	Enumeration Literal	SeverityKind	<code>\$SeverityKind[i].info</code>
warning	Enumeration Literal	SeverityKind	<code>\$SeverityKind[i].warning</code>

831. SharedProperty

A shared property specifies part of its containing block. Every shared property has 'shared' AggregationKind and is typed by a block. A shared property will be displayed under the 'references' compartment when 'Sort by SysML Style' in the 'Presentation Options' is selected.

Base Classifier

- [AbstractReferenceProperty](#)

832. Skill

MODAF: A specific set of abilities defined by knowledge, skills and attitude (Competence).

DoDAF: The ability, coming from one's knowledge, practice, aptitude, etc., to do something well.

Base Classifier

- [Competence](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Skill[i].actualPropertySet</code>
appliesTo	UPDMElement	PropertySet	<code>\$Skill[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$Skill[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Skill[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$Skill[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Skill[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Skill[i].URI</code>

833. SkillOfPersonType

Base Classifier

- [ProvidesCompetence](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SkillOfPersonType[i].actualPropertySet</code>

conformsTo	Standard	UPDMElement	\$\$SkillOfPersonType[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$\$SkillOfPersonType[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$\$SkillOfPersonType[i].propertySet
ProvidesCompetence.client		ProvidesCompetence	\$\$SkillOfPersonType[i].ProvidesCompetence.client
ProvidesCompetence.supplier		ProvidesCompetence	\$\$SkillOfPersonType[i].ProvidesCompetence.supplier
startBoundaryType	ISO8601DateTime	UPDMElement	\$\$SkillOfPersonType[i].startBoundaryType
universalPropertySet	ActualPropertySet	ProvidesCompetence	\$\$SkillOfPersonType[i].universalPropertySet
URI	String	UPDMElement	\$\$SkillOfPersonType[i].URI

834. SmartPackage

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
additionalElements	Element	SmartPackage	\$\$SmartPackage[i].additionalElements
excludedElements	Element	SmartPackage	\$\$SmartPackage[i].excludedElements
query	StructuredExpression	SmartPackage	\$\$SmartPackage[i].query

835. SOA Matrix

The Service Channel Summary Matrix provides summary of service channels between service providers and requesters (consumers).

It is editable matrix where cells of the matrix represent Service Channels and the headers represent System Resources.

To build the Matrix:

- 1. Specify Rows** scope (System Resources);
- 2. Specify Columns** scope (System Resources);
- 3. Click "Refresh"** button.

System Resources provides and consumes services through Service and Request ports. The Matrix is collecting and showing relationships between these ports called "Service Channels".

*By clicking on the cell context menu opens allowing to open new **Service Channel Creation wizard** or delete existing relationships.*

Base Classifier

- [InvisibleStereotype](#)

836. SOA Summary

The Service Channel Summary Table provides details of exchange elements being exchanged between service channels.

To fill in Service Channel Summary table, Existing Service Channels have to be added to it or new ones should be created.

*Click **Add New** button to create new Service Channel using **New Service Channel** creation wizard.*

*Click **Add Existing** button to select existing Service Channels.*

"Service Channel Name", "Provider's Service Interface", "Requester's Service Interface" and "Service Contract" cells are allowed to edit in the table. "Provided Items", "Service Provided Interfaces", "Service Required Interfaces" and "Service Exposed Capabilities" cells are allowed to edit if Provider's Service Interface is specified. "Required Items", "Request Provided Interfaces", "Request Required Interfaces" and "Request Exposed Capabilities" cells are allowed to edit if Requester's Service Interface is specified. All other cells are read only.

Rows (Service Channels) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SOA Summary	<code>§SOASummary[i].hideColumns</code>

837. Software

MODAF: An executable computer programme.

DoDAF: Materiel: Equipment, apparatus or supplies that are of interest, without distinction as to its application for administrative or combat purposes.

Base Classifier

- ResourceArtifact

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	\$Software[i].actsUpon
actualPropertySet	ActualPropertySet	UPDMElement	\$Software[i].actualPropertySet
affectedFunctions	Function	ResourceInteractionItem	\$Software[i].affectedFunctions
appliesTo	UPDMElement	PropertySet	\$Software[i].appliesTo
conformsTo	Standard	UPDMElement	\$Software[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$Software[i].endBoundaryType
milestone	ActualProjectMilestone	SystemResource	\$Software[i].milestone
physicalLocation	ActualLocation	LocationHolder	\$Software[i].physicalLocation
propertySet	PropertySet	UPDMElement	\$Software[i].propertySet
requiredEnvironment	Environment	LocationHolder	\$Software[i].requiredEnvironment
Resource.ownedOperation		SystemResource	\$Software[i].Resource.ownedOperation
Resource.ownedPort		SystemResource	\$Software[i].Resource.ownedPort
Resource.performs		SystemResource	\$Software[i].Resource.performs
startBoundaryType	ISO8601DateTime	UPDMElement	\$Software[i].startBoundaryType
URI	String	UPDMElement	\$Software[i].URI

838. SoftwareConceptRole

Base Classifier

- ConceptRole

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$SoftwareConceptRole[i].actualPropertySet
ConceptRole.type		ConceptRole	\$SoftwareConceptRole[i].ConceptRole.type
conformsTo	Standard	UPDMElement	\$SoftwareConceptRole[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$SoftwareConceptRole[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$SoftwareConceptRole[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$SoftwareConceptRole[i].startBoundaryType
URI	String	UPDMElement	\$SoftwareConceptRole[i].URI

839. SortingMode

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
----------------	----------------	-----------------	----------------------------------

Ascending	Enumeration Literal	SortingMode	<code>\$SortingMode[i].Ascending</code>
Custom	Enumeration Literal	SortingMode	<code>\$SortingMode[i].Custom</code>
Descending	Enumeration Literal	SortingMode	<code>\$SortingMode[i].Descending</code>

840. Source

Denotes a source file that can be compiled into an executable file. Subclass of <<file>>.

Base Classifier

- [File](#)

841. SOV-1

Base Classifier

- [InvisibleStereotype](#)

842. SOV-1 Package

Base Classifier

- [InvisibleStereotype](#)

843. SOV-2

Base Classifier

- [InvisibleStereotype](#)

844. SOV-2 Package

Base Classifier

- [InvisibleStereotype](#)

845. SOV-2 Report

Base Classifier

- [InvisibleStereotype](#)

846. SOV-3

The Capability to Service Mapping View (SOV-3) depicts which Service Interfaces contribute to the achievement of a Capability.

The Rows of this matrix are Service Interfaces and the Columns are Capabilities.

To build the Matrix:

- 1. Specify Rows scope (Service Interfaces);*
- 2. Specify Columns scope (Capabilities);*
- 3. Click "Refresh" button.*

Service Interfaces expose Capabilities using "Expose" relationship.

To map Service Interface to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

847. SOV-3 Package**Base Classifier**

- [InvisibleStereotype](#)

848. SOV-4a

Service Constraints (SOV-4a) table specifies constraints that apply to implementations of services.

There are two ways to fill this table.

- 1. Add new Service Policy. Click **Add New** button and select constrained Service Interface. Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*

2. Add Existing Service Policy. Click **Add Existing** button and select **Service Policies**.

Additional constrained elements can be added and edited for every Row in the table.

Rows (Service Policy) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SOV-4a	<code>\$(SOV-4a[i]).hideColumns</code>

849. SOV-4a Package

Base Classifier

- [InvisibleStereotype](#)

850. SOV-4a Report

Base Classifier

- [InvisibleStereotype](#)

851. SOV-4b

Base Classifier

- [InvisibleStereotype](#)

852. SOV-4b Package

Base Classifier

- [InvisibleStereotype](#)

853. SOV-4c

Base Classifier

- [InvisibleStereotype](#)

854. SOV-4c Package

Base Classifier

- [InvisibleStereotype](#)

855. SOV-5

Base Classifier

- [InvisibleStereotype](#)

856. SOV-5 Package

Base Classifier

- [InvisibleStereotype](#)

857. SpecialHandlingInstructions

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
approved for public release	Enumeration Literal	SpecialHandlingInstructions	<code>\$SpecialHandlingInstructions[i].approved for public release</code>
contingency assignment	Enumeration Literal	SpecialHandlingInstructions	<code>\$SpecialHandlingInstructions[i].contingency assignment</code>
critical nuclear weapon design information	Enumeration Literal	SpecialHandlingInstructions	<code>\$SpecialHandlingInstructions[i].critical nuclear weapon design information</code>
dissemination and extraction of information controlled by originator	Enumeration Literal	SpecialHandlingInstructions	<code>\$SpecialHandlingInstructions[i].dissemination and extraction of information controlled by originator</code>

not known	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].not known</code>
not releasable outside the US government	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].not releasable outside the US government</code>
not releasable to contractor or consultants	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].not releasable to contractor or consultants</code>
not releasable to foreign nationals	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].not releasable to foreign nationals</code>
not specified	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].not specified</code>
permanent assignment	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].permanent assignment</code>
proprietary information involved	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].proprietary information involved</code>
releasable to NATO	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].releasable to NATO</code>
releasable to SOIL country and NATO	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].releasable to SOIL country and NATO</code>
releasable to SOIL country only	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].releasable to SOIL country only</code>
sensitive compartmented information	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].sensitive compartmented information</code>
special access required	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].special access required</code>
special category	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].special category</code>
warning - intelligence sources or methods	Enumeration Literal	SpecialHandlingInstructions	<code>\$\$SpecialHandlingInstructions[i].warning - intelligence sources or methods</code>

858. Specification

A classifier that specifies a domain of objects without defining the physical implementation of those objects. For example, a Component stereotyped by «specification» will only have provided and required interfaces, and is not intended to have any realizingClassifiers as part of its definition. This differs from «type» because a «type» can have features such as attributes and methods which is useful to analysts modeling systems. Also see: «realization»

859. Stakeholder

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
concern	Comment	Stakeholder	<code>\$Stakeholder[i].concern</code>
concernList	Comment	Stakeholder	<code>\$Stakeholder[i].concernList</code>

860. Standard

MODAF: A ratified and peer-reviewed specification that is used to guide or constrain the architecture. A Standard may be applied to any element in the architecture via the [constrainedItem] property of UML::Constraint.

DoDAF: A formal agreement documenting generally accepted specifications or criteria for products, processes, procedures, policies, systems, and/or personnel.

Base Classifier

- [SubjectOfForecast](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Standard[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Standard[i].conformsTo</code>
currentStatus	String	Standard	<code>\$Standard[i].currentStatus</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Standard[i].endBoundaryType</code>
InformationTechnologyStandardCategory	String	Standard	<code>\$Standard[i].InformationTechnologyStandardCategory</code>
mandatedDate	ISO8601DateTime	Standard	<code>\$Standard[i].mandatedDate</code>
propertySet	PropertySet	UPDMElement	<code>\$Standard[i].propertySet</code>
ratifiedBy	ActualOrganization	Standard	<code>\$Standard[i].ratifiedBy</code>
retiredDate	ISO8601DateTime	Standard	<code>\$Standard[i].retiredDate</code>
shortName	String	Standard	<code>\$Standard[i].shortName</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Standard[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Standard[i].URI</code>
version	String	Standard	<code>\$Standard[i].version</code>

861. StandardConfiguration

MODAF: A UML::Comment that when attached to a CapabilityConfiguration indicates that it is a standard pattern for re-use in the architecture.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$StandardConfiguration[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$StandardConfiguration[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$StandardConfiguration[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$StandardConfiguration[i].propertySet</code>
StandardConfiguration.annotatedElement		StandardConfiguration	<code>\$StandardConfiguration[i].StandardConfiguration.annotatedElement</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$StandardConfiguration[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$StandardConfiguration[i].URI</code>

862. StandardOperationalActivity

MODAF: An OperationalActivity that is a standard procedure that is doctrinal . Note: This is equivalent to what some defence organisations call JETLs.

DoDAF: Work, not specific to a single organization, weapon system or individual, that transforms inputs into outputs or changes their state (DoDAF:: Activity).

Base Classifier

- [OperationalActivity](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
activityPerformableUnderCondition	Environment	Activity	<code>\$StandardOperationalActivity[i].activityPerformableUnderCondition</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$StandardOperationalActivity[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$StandardOperationalActivity[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$StandardOperationalActivity[i].endBoundaryType</code>
OperationalActivity.ownedParameter		OperationalActivity	<code>\$StandardOperationalActivity[i].OperationalActivity.ownedParameter</code>
propertySet	PropertySet	UPDMElement	<code>\$StandardOperationalActivity[i].propertySet</code>
realizedBy	NodeOperation	OperationalActivity	<code>\$StandardOperationalActivity[i].realizedBy</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$StandardOperationalActivity[i].startBoundaryType</code>
subject	ActivitySubject	OperationalActivity	<code>\$StandardOperationalActivity[i].subject</code>
URI	String	UPDMElement	<code>\$StandardOperationalActivity[i].URI</code>

863. Standards Viewpoint

Base Classifier

- [InvisibleStereotype](#)

864. StatusIndicators

UPDM: Specifies a status for a ProjectTheme (such as training status).

MODAF: An enumeration of the possible statuses (MODAF::StatusIndicator) for one or more ProjectThemes.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$StatusIndicators[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$StatusIndicators[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$StatusIndicators[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$StatusIndicators[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$StatusIndicators[i].startBoundaryType
URI	String	UPDMElement	\$StatusIndicators[i].URI

865. StdV-1

Standards Profile (StdV-1) table defines listing of standards that apply to solution elements.

There are two ways to add a row in this table:

1. Add new UPDM Element. Click “**Add new UPDM Element**” button. Select UPDM Element (this term means any of the UPDM available element). Specify the owner for the selected element.

2. Add Existing UPDM elements. Click “**Add Existing UPDM Element**” button and select UPDM elements.

For each row there should be assigned one or more Standards or Protocols that the particular UPDM element must conform to. Click “...” button on any cell in the "Standard/Policy" column.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

866. StdV-1 Package

Base Classifier

- [InvisibleStereotype](#)

867. StdV-2

Standards Forecast (StdV-2) table defines emerging standards and potential impact on current solution elements, within a set of time frames.

You will find this table identical to the SV-9 table. In general they are identical in implementation, but StdV-2 is more likely to be used for Standards and Protocols forecasting.

Three major steps should be done to create the table:

1. Add Rows to the Table. There are two ways to add a row to the table:

1.1. Add new Subject of Forecast as row Header. Click "Create New Subject of Forecast..." button and select element you want to create. Specify owner for selected element.

1.2. Add Existing subject of forecast as row Header. Click "Add Existing Subject of Forecast" button and select one or more existing elements.

2. Add columns to the table. Click "Add/Remove forecast" button. Specify Time Periods for the forecasting: select or create Time Line Package to store Forecast dates; select forecast kind. According to the selected Forecast Kind, specify additional options needed (see Forecasting Period Dialog for more info).

3. Fill in the cells with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select System Resources, Functional or Technical Standards, Protocols or Skills to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected Time Period.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

868. StdV-2 Package

Base Classifier

- [InvisibleStereotype](#)

869. StereotypeExtension

MODAF: Defines an additional stereotype used in the architecture which is not defined in this meta-model. The body attribute contains the name of the new stereotype. The extendedStereotype tagged value shall contain the name of the meta-model stereotype which is extended. The ontologyReference tagged value shall be populated with a reference to the external ontology element represented by the new stereotype.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$\$StereotypeExtension[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$\$StereotypeExtension[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$\$StereotypeExtension[i].endBoundaryType</code>
ontologyReference	OntologyReference	StereotypeExtension	<code>\$\$StereotypeExtension[i].ontologyReference</code>
propertySet	PropertySet	UPDMElement	<code>\$\$StereotypeExtension[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$\$StereotypeExtension[i].startBoundaryType</code>
StereotypeExtension.annotatedElement		StereotypeExtension	<code>\$\$StereotypeExtension[i].StereotypeExtension.annotatedElement</code>
URI	String	UPDMElement	<code>\$\$StereotypeExtension[i].URI</code>

870. StereotypesMappingRule

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
DoDAF	Stereotype	StereotypesMappingRule	<code>\$\$StereotypesMappingRule[i].DoDAF</code>
DoDAF_2_0	Stereotype	StereotypesMappingRule	<code>\$\$StereotypesMappingRule[i].DoDAF_2_0</code>
MODAF	Stereotype	StereotypesMappingRule	<code>\$\$StereotypesMappingRule[i].MODAF</code>
NAF	Stereotype	StereotypesMappingRule	<code>\$\$StereotypesMappingRule[i].NAF</code>
NAF_4_0	Stereotype	StereotypesMappingRule	<code>\$\$StereotypesMappingRule[i].NAF_4_0</code>
Prior	Boolean	StereotypesMappingRule	<code>\$\$StereotypesMappingRule[i].Prior</code>

871. Strategic Viewpoint

Base Classifier

- [InvisibleStereotype](#)

872. streaming

Used for activities that can accept inputs or provide outputs after they start and before they finish.

873. String**874. StringsMappingRule**

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
DoDAF_2_0_String	String	StringsMappingRule	<code>StringsMappingRule[i].DoDAF_2_0_String</code>
DoDAF_String	String	StringsMappingRule	<code>StringsMappingRule[i].DoDAF_String</code>
MODAF_String	String	StringsMappingRule	<code>StringsMappingRule[i].MODAF_String</code>
NAF_4_0_String	String	StringsMappingRule	<code>StringsMappingRule[i].NAF_4_0_String</code>
NAF_String	String	StringsMappingRule	<code>StringsMappingRule[i].NAF_String</code>
Prior	Boolean	StringsMappingRule	<code>StringsMappingRule[i].Prior</code>

875. StructuralPart

UPDM: An *EnterprisePhase* can be sub-divided into structural and temporal parts. *StructuralPart* describes the *EnterprisePhase* elements that describe the structure.
 MODAF: Asserts that one *EnterprisePhase* is a spatial part of another, (*MODAF::EnterpriseStructure*) Note:- This is a topological structuring relationship, hence the *EnterprisePhase* may be physically disjoint

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>StructuralPart[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>StructuralPart[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>StructuralPart[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>StructuralPart[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>StructuralPart[i].startBoundaryType</code>
StructuralPart.class		StructuralPart	<code>StructuralPart[i].StructuralPart.class</code>
StructuralPart.type		StructuralPart	<code>StructuralPart[i].StructuralPart.type</code>

URI	String	UPDMElement	<code>\$StructuralPart[i].URI</code>
-----	--------	-----------------------------	--------------------------------------

876. StV-1

Base Classifier

- [InvisibleStereotype](#)

877. StV-1 Package

Base Classifier

- [InvisibleStereotype](#)

878. StV-2

Base Classifier

- [InvisibleStereotype](#)

879. StV-2 Package

Base Classifier

- [InvisibleStereotype](#)

880. StV-3

Base Classifier

- [InvisibleStereotype](#)

881. StV-3 Package

Base Classifier

- [InvisibleStereotype](#)

882. StV-4

Base Classifier

- [InvisibleStereotype](#)

883. StV-4 Package

Base Classifier

- [InvisibleStereotype](#)

884. StV-5

StV-5 Capability to Organisation Deployment Mapping (StV-5) shows deployment of capability configurations to specific organizations during a specific Enterprise Phase.

*Click **Add Rows** button to select Actual Organizational Resources. Selected Actual Organizational Resources shall be added as table rows.*

*Click **Add/Remove Columns** button to select or deselect Capabilities displayed as Columns in the table.*

Table cells are allowed to edit in the table. System Resources can be added to the table as cells contents. Added resources will be marked as exhibiting Capability represented by column and used by Actual Organizational Resource represented by Row in the particular Enterprise time frame.

Rows (Actual Organization Resources) can be removed from model or only from table, can be ordered, exported to the CSV or HTML and StV-5 Spreadsheet Report can be generated.

More Actions are available by clicking right mouse button on a cell

Base Classifier

- [InvisibleStereotype](#)

885. StV-5 Package

Base Classifier

- [InvisibleStereotype](#)

886. StV-5 Report**Base Classifier**

- [InvisibleStereotype](#)

887. StV-6

The Operational Activity to Capability Mapping (StV-6) describes the mapping between the capabilities required by an Enterprise and the operational activities that those capabilities support.

The Rows of this matrix are Capabilities and the Columns are Standard Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Capabilities);*
- 2. Specify Columns scope (Standard Operational Activities);*
- 3. Click "Refresh" button.*

Standard Operational Activities maps to Capabilities using "Maps to Capability" relationship.

To map Standard Operational Activity to Capability, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

888. StV-6 Package**Base Classifier**

- [InvisibleStereotype](#)

889. SubcontentsKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
All	Enumeration Literal	SubcontentsKind	<code>\$SubcontentsKind[i].All</code>
None	Enumeration Literal	SubcontentsKind	<code>\$SubcontentsKind[i].None</code>
Some	Enumeration Literal	SubcontentsKind	<code>\$SubcontentsKind[i].Some</code>

890. SubjectOfForecast

MODAF: Abstract Any element that may be subject to a Forecast.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SubjectOfForecast[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$SubjectOfForecast[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SubjectOfForecast[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$SubjectOfForecast[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SubjectOfForecast[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$SubjectOfForecast[i].URI</code>

891. SubjectOfOperationalConstraint

MODAF: Abstract. An element of the architecture that may be subject to an OperationalConstraint or OperationalStateDescription.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SubjectOfOperationalConstraint[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$SubjectOfOperationalConstraint[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SubjectOfOperationalConstraint[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$SubjectOfOperationalConstraint[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SubjectOfOperationalConstraint[i].startBoun</code>

			daryType
URI	String	UPDMElement	\$SubjectOfOperationalConstraint[i].URI

892. SubjectOfOperationalStateMachine

UPDM Abstract Element: The element being described by the state machine.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$SubjectOfOperationalStateMachine[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$SubjectOfOperationalStateMachine[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$SubjectOfOperationalStateMachine[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$SubjectOfOperationalStateMachine[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$SubjectOfOperationalStateMachine[i].startBoundaryType
SubjectOfOperationalStateMachine.ownedBehavior		SubjectOfOperationalStateMachine	\$SubjectOfOperationalStateMachine[i].SubjectOfOperationalStateMachine.ownedBehavior
URI	String	UPDMElement	\$SubjectOfOperationalStateMachine[i].URI

893. SubjectOfResourceConstraint

MODAF: Abstract. Anything that may be constrained by a ResourceConstraint.

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$SubjectOfResourceConstraint[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$SubjectOfResourceConstraint[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$SubjectOfResourceConstraint[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$SubjectOfResourceConstraint[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$SubjectOfResourceConstraint[i].startBoundaryType

URI	String	UPDMElement	\$SubjectOfResourceConstraint[i].URI
-----	--------	-----------------------------	--------------------------------------

894. Subsystem

A Subsystem is a - typically large - encapsulated block within a larger system.

Base Classifier

- [Block](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	\$Subsystem[i].isEncapsulated

895. Subsystem

A unit of hierarchical decomposition for large systems. A subsystem is commonly instantiated indirectly. Definitions of subsystems vary widely among domains and methods, and it is expected that domain and method profiles will specialize this construct. A subsystem may be defined to have specification and realization elements. See also: «specification» and «realization».

896. suggestedStringValues

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
values	String	suggestedStringValues	\$suggestedStringValues[i].values

897. suggestedValues

898. SV-1

Base Classifier

- [InvisibleStereotype](#)

899. SV-1 Package

Base Classifier

- [InvisibleStereotype](#)

900. SV-1 Package MODAF

Base Classifier

- [InvisibleStereotype](#)

901. SV-10a

Resource Constraints Specification (SV-10a) allows you to constraint Systems View Architectural elements.

There are two ways to fill this table:

- 1. Add new Resource Constraint. Click **Add New** button and select Systems Element (Resource Artifact, Software, Capability Configuration, Organization, Post, Function, Exchange Element, Entity Item, Resource Interaction) to be constrained. Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. Add Existing Resource Constraints. Click **Add Existing** button and select Resource Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Resource Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-10a	<code>\$SV-10a[i].hideColumns</code>

902. SV-10a Package

Base Classifier

- [InvisibleStereotype](#)

903. SV-10a Report**Base Classifier**

- [InvisibleStereotype](#)

904. SV-10a-DoDAF2

Systems Rules Model (SV-10a) identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.

There are two ways to fill this table:

*1. Add new Resource Constraint. Click **Add New** button and select constrained Systems Element (System, Software, Capability Configuration, Organization Type, Person Type, Service Access, Function, Exchange Element, Entity Item, Resource Interaction) to be constrained. Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*

*2. Add Existing Resource Constraints. Click **Add Existing** button and select Resource Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Resource Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-10a-DoDAF2	\$SV-10a-DoDAF2 [i] .hideColumns

905. SV-10b**Base Classifier**

- [InvisibleStereotype](#)

906. SV-10b Package

Base Classifier

- [InvisibleStereotype](#)

907. SV-10c

Base Classifier

- [InvisibleStereotype](#)

908. SV-10c Package

Base Classifier

- [InvisibleStereotype](#)

909. SV-11

Base Classifier

- [InvisibleStereotype](#)

910. SV-11 Package

Base Classifier

- [InvisibleStereotype](#)

911. SV-11 Package MODAF

Base Classifier

- [InvisibleStereotype](#)

912. SV-12

Service Provision (SV-12) Matrix defines the relationships between the Resources and Service Interfaces.

The Rows of this matrix are Service Interfaces and the Columns are System Resources.

To build the Matrix:

- 1. Specify Rows scope (Service Interfaces);*
- 2. Specify Columns scope (System Resources);*
- 3. Click "Refresh" button.*

Cells here represents Services or Requests. If the resource provides service, Service Icon is displayed. If it requests (consumes) service, Request Icon is displayed.

By double clicking on the empty cell new Service will be created. By double clicking once more new Request will be created and the Service will be deleted. By double clicking third time cell will be cleared and the Request will be deleted.

Base Classifier

- InvisibleStereotype

913. SV-12 Package

Base Classifier

- InvisibleStereotype

914. SV-13

The Technology & Skills Forecast (SV-9) defines the underlying current and expected supporting technologies and skills.

Three major steps should be done to create the table:

1. Add Rows to the Table. There are two ways to add a row to this table:

1.1. Add new Resource as row Header. Click “**Create New Resource...**” button and select Resource you want to create. Specify owner for selected Resource.

1.2. Add Existing Resource as row Header. Click “**Add Existing Resource...**” button and select one or more existing Resources.

2. Add columns to the table. Click “**Add/Remove forecast**” button. Specify Time Periods for the forecasting: select or create Time Line Package to store Forecast dates; select forecast kind. According to the selected Forecast Kind, specify additional options needed (see Forecasting Period Dialog for more info).

3. Fill in the cells with the Subjects of Forecast. Click the “...” button on the cell you want to fill in, select Resources, Standards, Protocols or Competences to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected Time Period.

Rows (Resources) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

915. SV-1i

Base Classifier

- [InvisibleStereotype](#)

916. SV-2

Base Classifier

- [InvisibleStereotype](#)

917. SV-2 Package

Base Classifier

- [InvisibleStereotype](#)

918. SV-2 Package MODAF

Base Classifier

- [InvisibleStereotype](#)

919. SV-2i

Base Classifier

- [InvisibleStereotype](#)

920. SV-3

Resource Interaction Matrix (SV-3) shows how resources interact to each other.

It is editable matrix where cells of the matrix represent Resource interactions and the headers represent System Resources.

Arrow can be on one side and on the other side. It depends on the direction of Resource interaction flow between System Resources. SV-3 matrix consists of as much rows and columns as there are System Resources in the matrix data source.

To build the Matrix:

- 1. Specify Rows scope (System Resources);*
- 2. Specify Columns scope (System Resources);*
- 3. Click "Refresh" button.*

By clicking on the cell context menu opens allowing to open new Resource Interaction creation wizard or delete existing relationships.

Base Classifier

- [InvisibleStereotype](#)

921. SV-3 Package

Base Classifier

- [InvisibleStereotype](#)

922. SV-3-DoDAF2

The Systems-Systems Matrix (SV-3) shows how systems interact to each other.

It is editable matrix where cells of the matrix represent Resource interactions and the headers represent System Resources.

Arrow can be on one side and on the other side. It depends on the direction of Resource interaction flow between System Resources. SV-3 matrix consists of as much rows and columns as there are System Resources in the matrix data source.

To build the Matrix:

- 1. Specify Rows scope (System Resources);*
- 2. Specify Columns scope (System Resources);*
- 3. Click "Refresh" button.*

By clicking on the cell context menu opens allowing to open new Resource Interaction creation wizard or delete existing relationships.

Base Classifier

- [InvisibleStereotype](#)

923. SV-4**Base Classifier**

- [InvisibleStereotype](#)

924. SV-4 Package**Base Classifier**

- [InvisibleStereotype](#)

925. SV-4 Package MODAF

Base Classifier

- [InvisibleStereotype](#)

926. SV-5

The Function to Operational Activity Traceability Matrix (SV-5) addresses the linkage between Functions described in SV-4 and Operational Activities specified in OV-5.

The Rows of this matrix are Functions and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Functions);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Functions implements Operational Activities using "Implements" relationship.

To map Function to Operational Activity, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- [InvisibleStereotype](#)

927. SV-5 Package

Base Classifier

- [InvisibleStereotype](#)

928. SV-5a

Operational Activity to Systems Function Traceability Matrix (SV-5a) addresses the linkage between Functions described in SV-4 and Operational Activities specified in OV-5.

The Rows of this matrix are Functions and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Functions);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Functions implements Operational Activities using "Implements" relationship.

To map Function to Operational Activity, double click on the intersection between the desired elements. By double clicking on the intersection again, the relation will be deleted.

Base Classifier

- InvisibleStereotype

929. SV-5b

The Operational Activity to Systems Traceability Matrix (SV-5b) describes the mapping between the Systems and Operational Activities.

The Rows of this matrix are Systems and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Systems);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Systems maps to Operational Activities through transitive relationship. It consists of mapping between Operational Activities and Functions using Implements relationship and mapping between Systems and Functions using Activity Performed by Performer relationship.

Base Classifier

- [InvisibleStereotype](#)

930. SV-6

The Systems Data Exchange Matrix (SV-6) specifies the characteristics of the data exchanged between Resources.

To fill in SV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add New** button to create a new Resource Interaction between selected System Resources.*

*Click **Add Existing** button and select Resource Interactions or Resource Interfaces. In case Resource Interface is selected, all Resource Interactions flowing via it will be added to the table.*

"Resource Interaction identifier". "Resource Interaction Item Name, Producing and Consuming Functions, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Resource Interactions SV-1 product is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-6	\$SV-6[i].hideColumns

931. SV-6 Package**Base Classifier**

- [InvisibleStereotype](#)

932. SV-6 Report

Base Classifier

- [InvisibleStereotype](#)

933. SV-6 Role Based

The Role-based System Data Exchange Matrix (SV-6)

addresses specifies the characteristics of the data exchanged between resources.

To fill in SV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add Existing** button to select Resource Interactions, Resource Interfaces, Resource Connectors, or Service Channels. In case Resource Interface, Resource Connector, or Service Channel is selected, all Resource Interactions flowing via it will be added to the table.*

"Interaction ID", "Resource Interaction Item", "Producing and Consuming Functions", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Resource Interactions SV-1 internal diagram is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be filtered according to the connector kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-6 Role Based	<code>\$SV-6RoleBased[i].hideColumns</code>

934. SV-6 Role Based Report

Base Classifier

- [InvisibleStereotype](#)

935. SV-6-DoDAF2

The Systems Resource Flow Matrix (SV-6) provides details of system resource flow elements being exchanged between systems and the attributes of that exchange.

To fill in SV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add New** button to create a new Resource Interaction between selected System Resources.*

*Click **Add Existing** button and select Resource Interactions or Resource Interfaces. In case Resource Interface is selected, all Resource Interactions flowing via it will be added to the table.*

"Resource Interaction identifier". "Resource Interaction Item Name, Producing and Consuming Functions, and wide range of measurements" cells are allowed to edit in the table. All other cells are read only. For creation and modification of Resource Interactions SV-1 product is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-6-DoDAF2	<code>\$SV-6-DoDAF2[i].hideColumns</code>

936. SV-6-DoDAF2 Role Based

The Role-based Systems Resource Flow Matrix (SV-6)

addresses specifies the characteristics of the data exchanged between resources.

To fill in SV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add Existing** button to select Resource Interactions, Resource Interfaces, Resource Connectors, or Service Channels. In case Resource Interface, Resource Connector, or Service Channel is selected, all Resource Interactions flowing via it will be added to the table.*

"Interaction ID", "Resource Interaction Item", "Producing and Consuming Functions", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Resource Interactions SV-1 internal diagram is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be filtered according to the connector kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-6-DoDAF2 Role Based	\$SV-6-DoDAF2RoleBased[i].hideColumns

937. SV-7 Actual

The Resource Performance Actual Parameters Matrix (SV-7 Actual) depicts the Actual values of performance characteristics of a Resource.

There are three ways to add a row in this table:

1. **Add new** measurable Resource; Click **Add New** button and select one or more System Resources that have at least one Measurement Set Defined (see SV-7 Typical). Specify values for each Measurement - directly in the table cells.
2. **Add existing** Measures or measurable Resources. Click **Add Existing** button and select Existing Measurements or Resources.
3. **Add missing** Measurements. Click "**Add the missing Measurements**" button to update table to include latest model changes.

Rows (Actual Measurements) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-7 Actual	\$SV-7Actual[i].hideColumns

938. SV-7 Actual-DoDAF2

The Systems Actual Measures Matrix (SV-7 Actual) depicts the Actual values of performance characteristics of a System.

There are three ways to add a row in this table:

1. **Add new measurable Resource**; Click **Add New** button and select one or more System Resources that have at least one Measure Type defined (see SV-7 Typical). Specify values for each Measurement - directly in the table cells.
2. **Add existing Measures or measurable Resources**. Click **Add Existing** button and select Existing Measurements or System Resources.
3. **Add missing Measurements**. Click “**Add the missing Measurements**” button to update table to include latest model changes.

Rows (Actual Measurements) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-7 Actual-DoDAF2	\$SV-7Actual-DoDAF2[i].hideColumns

939. SV-7 Package

Base Classifier

- [InvisibleStereotype](#)

940. SV-7 Report

Base Classifier

- [InvisibleStereotype](#)

941. SV-7 Typical

The Resource Performance Typical Parameters Matrix (SV-7 Typical) is the and depicts the possible types of performance characteristics of a Resource.

There are two ways to add a row in this table:

1. **Add new Measurement Set**. Click **Add New** button and select the owning element for Measurement Set. Specify Measurements to the Measurement Set and Resources to be Measured - straight in the table cells.
2. **Add existing Measurements Sets**. Click **Add Existing** button and select Existing Measurement Sets.

Rows (*Measurement Sets*) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-7 Typical	\$SV-7Typical[i].hideColumns

942. SV-7 Typical-DoDAF2

The Systems Typical Measures Matrix (SV-7 Typical) depicts the possible types of performance characteristics of a System.

There are two ways to add a row in this table:

1. **Add new Measure Type.** Click **Add New** button and select the owning element for Measure Type. Specify Measurements to the Measure Type and Resources to be Measured - directly in the table cells.
2. **Add existing Measure Types.** Click **Add Existing** button and select Existing Measure Types.

Rows (*Measure Types*) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SV-7 Typical-DoDAF2	\$SV-7Typical-DoDAF2[i].hideColumns

943. SV-8

Base Classifier

- [InvisibleStereotype](#)

944. SV-8 Package

Base Classifier

- [InvisibleStereotype](#)

945. SV-8 Report

Base Classifier

- [InvisibleStereotype](#)

946. SV-8-DoDAF2

Base Classifier

- [InvisibleStereotype](#)

947. SV-9

The Technology & Skills Forecast (SV-9) defines the underlying current and expected supporting technologies and skills.

Three major steps should be done to create the table:

1. Add Rows to the Table. There are two ways to add a row to this table:

*1.1. Add new Resource as row Header. Click **Add New** button and select System Resource you want to create. Specify owner for selected Resource.*

*1.2. Add Existing Resource as row Header. Click **Add Existing** button and select one or more existing System Resources.*

*2. Add columns to the table. Click "**Add/Remove forecast**" button. Specify time periods for the forecasting: select or create Time Line Package to store forecast dates; select forecast kind. According to the selected forecast kind, specify additional options needed (see Forecasting Period Dialog help for more information).*

3. Fill in the cells with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select System Resources, Standards, Protocols or Competences to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected time period.

Rows (System Resources) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

948. SV-9 Package

Base Classifier

- [InvisibleStereotype](#)

949. SV-9-DoDAF2

Systems Technology & Skills Forecast (SV-9) defines emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future system development.

Three major steps should be done to create the table:

1. Add Rows to the Table. There are two ways to add a row to this table:

*1.1. Add new Subject of Forecast as row Header. Click **Add New** button and select element you want to create. Specify owner for selected element.*

*1.2. Add Existing Subject of Forecast as row Header. Click **Add Existing** button and select one or more existing elements.*

*2. Add columns to the table. Click "**Add/Remove forecast**" button. Specify Time Periods for the forecasting: select or create Time Line package to store Forecast dates; select forecast kind. According to the selected Forecast Kind, specify additional options needed (see Forecasting Period Dialog for more info).*

3. Fill in the cells with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select System Resources, Standards, Protocols or Skills to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected time period.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

950. SvcV-1

Base Classifier

- [InvisibleStereotype](#)

951. SvcV-1 Package**Base Classifier**

- [InvisibleStereotype](#)

952. SvcV-10a

Services Rules Model (SvcV-10a) table identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.

There are two ways to fill this table:

- 1. **Add new Resource Constraint.** Click **Add New** button and select Service Access to be constrained. Fill in specification cell. It is an expression that can be written in natural or technical language (for example English or OCL).*
- 2. **Add Existing Resource Constraints.** Click **Add Existing** button and select Resource Constraints.*

Additional constrained elements can be added and edited for every Row in the table.

Rows (Resource Constraints) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SvcV-10a	<code>\$\$SvcV-10a[i].hideColumns</code>

953. SvcV-10a Package**Base Classifier**

- [InvisibleStereotype](#)

954. SvcV-10a Report

Base Classifier

- [InvisibleStereotype](#)

955. SvcV-10b

Base Classifier

- [InvisibleStereotype](#)

956. SvcV-10b Package

Base Classifier

- [InvisibleStereotype](#)

957. SvcV-10c

Base Classifier

- [InvisibleStereotype](#)

958. SvcV-10c Package

Base Classifier

- [InvisibleStereotype](#)

959. SvcV-2

Base Classifier

- [InvisibleStereotype](#)

960. SvcV-2 Package

Base Classifier

- [InvisibleStereotype](#)

961. SvcV-2i

Base Classifier

- [InvisibleStereotype](#)

962. SvcV-3a

Systems-Services Matrix (SvcV-3a) Matrix relationships among or between systems and services in a given Architectural Description.

It is editable matrix where cells of the matrix represent Resource interactions and the headers represent Service Accesses and System Resources.

Arrow can be on one side and on the other side. It depends on the direction of Resource interaction flow between System Resources.

To build the Matrix:

- 1. Specify **Rows** scope (Service Accesses);*
- 2. Specify **Columns** scope (System Resources);*
- 3. Click "**Refresh**" button.*

By clicking on the cell context menu opens allowing to open new Resource Interaction creation wizard or delete existing relationships.

Base Classifier

- [InvisibleStereotype](#)

963. SvcV-3a Package

Base Classifier

- [InvisibleStereotype](#)

964. SvcV-3b

Systems-Services Matrix (SvcV-3b) Matrix defines the relationships between the Systems and Service Interfaces.

It is editable matrix where cells of the matrix represent Resource interactions and the headers represent Service Accesses.

Arrow can be on one side and on the other side. It depends on the direction of Resource interaction flow between Service Accesses.

To build the Matrix:

- 1. Specify **Rows** scope (System Accesses);*
- 2. Specify **Columns** scope (System Accesses);*
- 3. Click "**Refresh**" button.*

By clicking on the cell context menu opens allowing to open new Resource Interaction creation wizard or delete existing relationships.

Base Classifier

- [InvisibleStereotype](#)

965. SvcV-3b Package

Base Classifier

- [InvisibleStereotype](#)

966. SvcV-4

Base Classifier

- [InvisibleStereotype](#)

967. SvcV-4 Package

Base Classifier

- [InvisibleStereotype](#)

968. SvcV-5

Operational Activity to Services Traceability Matrix (SvcV-5) describes the mapping of services back to operational activities

The Rows of this matrix are Service Accesses and the Columns are Operational Activities.

To build the Matrix:

- 1. Specify Rows scope (Service Accesses);*
- 2. Specify Columns scope (Operational Activities);*
- 3. Click "Refresh" button.*

Service Accesses maps to Operational Activities through transitive relationship. It consists of mapping between Operational Activities and Functions using Implements relationship and mapping between Service Accesses and Functions using Activity Performed by Performer relationship.

Base Classifier

- [InvisibleStereotype](#)

969. SvcV-5 Package**Base Classifier**

- [InvisibleStereotype](#)

970. SvcV-6

The SvcV-6 Services Resource Flow Matrix (SvcV-6) provides details of service Resource Flow elements being exchanged between services and the attributes of that exchange.

To fill in SvcV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add Existing** button to select Resource Interactions, Resource Interfaces, Resource Connectors, or Service Channels. In case Resource Interface, Resource Connector, or Service Channel is selected, all Resource Interactions flowing via it will be added to the table.*

"Interaction ID", "Resource Interaction Item", "Producing and Consuming Functions", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Resource Interactions SV-1 internal diagram is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be filtered according to the connector kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Rows (Resource Interactions) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SvcV-6	\$SvcV-6[i].hideColumns

971. SvcV-6 Package

Base Classifier

- [InvisibleStereotype](#)

972. SvcV-6 Report

Base Classifier

- [InvisibleStereotype](#)

973. SvcV-6 Role Based

The Role-based Services Resource Flow Matrix (SvcV-6) provides details of service Resource Flow elements being exchanged between services and the attributes of that exchange.

To fill in SvcV-6 table, Existing Resource Interactions have to be added to it.

*Click **Add Existing** button to select Resource Interactions, Resource Interfaces, Resource Connectors, or Service Channels. In case Resource Interface, Resource Connector, or Service Channel is selected, all Resource Interactions flowing via it will be added to the table.*

"Interaction ID", "Resource Interaction Item", "Producing and Consuming Functions", and wide range of measurement cells are allowed to edit in the table. For creation and modification of Resource Interactions SV-1 internal diagram is recommended to use.

Rows (Resource Interactions) can be removed from model or only from table, can be filtered according to the connector kind, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SvcV-6 Role Based	\$SvcV-6RoleBased[i].hideColumns

974. SvcV-6 Role Based Report

Base Classifier

- [InvisibleStereotype](#)

975. SvcV-7 Actual

The Services Actual Measures Matrix (SvcV-7 Actual) depicts the Actual values of performance characteristics of a Service.

There are three ways to add a row in this table:

- 1. Add new measurable Service; Click **Add New** button and select one or more Service Accesses that have at least one Measure Type defined (see SvcV-7 Typical). Specify values for each Measurement - directly in the table cells.*
- 2. Add existing Measures or measurable Services. Click **Add Existing** button and select Existing Measurements or Service Accesses.*
- 3. Add missing Measurements. Click "**Add the missing Measurements**" button to update table to include latest model changes.*

Rows (Actual Measurements) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SvcV-7 Actual	\$SvcV-7Actual[i].hideColumns

976. SvcV-7 Package

Base Classifier

- [InvisibleStereotype](#)

977. SvcV-7 Report

Base Classifier

- [InvisibleStereotype](#)

978. SvcV-7 Typical

The Services Typical Measures Matrix (SvcV-7 Typical) depicts the possible types of performance characteristics of a Service.

There are two ways to add a row in this table:

- 1. Add new Measure Type. Click **Add New** button and select the owning element for Measure Type. Specify Measurements to the Measure Type and Services to be Measured - directly in the table cells.*
- 2. Add existing Measure Types. Click **Add Existing** button and select Existing Measure Types.*

Rows (Measure Types) can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
hideColumns	String	SvcV-7 Typical	\$SvcV-7Typical[i].hideColumns

979. SvcV-8

Base Classifier

- [InvisibleStereotype](#)

980. SvcV-8 Package

Base Classifier

- [InvisibleStereotype](#)

981. SvcV-8 Report

Base Classifier

- [InvisibleStereotype](#)

982. SvcV-9

Services Technology & Skills Forecast (SvcV-9) table defines emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future service development.

You will find this table identical to the SV-9 table. In general they are identical in implementation, but SvcV-9 is more likely to be used for Services forecasting.

Three major steps should be done to create the table:

*1. **Add Rows** to the Table. There are two ways to add a row to this table:*

*1.1. **Add new** Subject of Forecast as row Header. Click **Add New** button and select element you want to create. Specify owner for selected element.*

*1.2. **Add Existing** Subject of Forecast as row Header. Click **Add Existing** button and select one or more existing elements.*

*2. **Add columns** to the table. Click "**Add/Remove forecast**" button. Specify Time Periods for the forecasting: select or create Time Line package to store Forecast dates; select forecast kind. According to the selected Forecast Kind, specify additional options needed (see Forecasting Period Dialog for more info).*

*3. **Fill in the cells** with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select System Accesses to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected time period.*

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

983. SvcV-9 Package

Base Classifier

- [InvisibleStereotype](#)

984. SwimLaneDiagram

Activity diagram usage with swim lanes.

985. System

A System is an artificial artifact consisting of blocks that pursue a common goal that cannot be achieved by the system's individual elements. A block can be software, hardware, a person, or an arbitrary unit.

Base Classifier

- [Block](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	<code>\$System[i].isEncapsulated</code>

986. System

A DoDAF alias for ResourceArtifact.

Base Classifier

- [ResourceArtifact](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$System[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$System[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$System[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$System[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$System[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$System[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$System[i].milestone</code>

physicalLocation	ActualLocation	LocationHolder	<code>\$System[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$System[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$System[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$System[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$System[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$System[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$System[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$System[i].URI</code>

987. System context

A *System context element* is a virtual container that includes the entire system and its actors.

Base Classifier

- [Block](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
isEncapsulated	Boolean	Block	<code>\$Systemcontext[i].isEncapsulated</code>

988. System process

989. System Resource Map

The **System Resource Map** depicts the structural decomposition of a **System Resource** including *Capability Configurations, Organization Types, Software and other elements*.

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:


Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node  / .

Move the whole structure - click on the empty place in the Relation Map and drag.

Move the selected Node - click on the Node and drag.

Insert New Element - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.

Insert New Element of the Same Type - type the name and press Ctrl+Enter to create an element of the same type.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- [InvisibleStereotype](#)

990. SystemAction

Base Classifier

- [InvisibleStereotype](#)

991. SystemConceptRole

Base Classifier

- [ConceptRole](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SystemConceptRole[i].actualPropertySet</code>
ConceptRole.type		ConceptRole	<code>\$SystemConceptRole[i].ConceptRole.type</code>
conformsTo	Standard	UPDMElement	<code>\$SystemConceptRole[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SystemConceptRole[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$SystemConceptRole[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SystemConceptRole[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$SystemConceptRole[i].URI</code>

992. SystemModel

A *systemModel* is a stereotyped model that contains a collection of models of the same physical system. A *systemModel* also contains all relationships and constraints between model elements contained in different models.

993. SystemResource

UPDM: Abstract supertype for physical resources such as *OrganizationalResource*.

MODAF: A *PhysicalAsset*, *OrganisationalResource* or *FunctionalResource* that can contribute towards fulfilling a capability (*MODAF::ResourceType*).

Base Classifier

- [OperationalExchangeItem](#)
- [Participant](#)
- [ResourceInteractionItem](#)
- [SubjectOfForecast](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actsUpon	OperationalActivity	ActivitySubject	<code>\$SystemResource[i].actsUpon</code>
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$SystemResource[i].actualPropertySet</code>
affectedFunctions	Function	ResourceInteractionItem	<code>\$SystemResource[i].affectedFunctions</code>
appliesTo	UPDMElement	PropertySet	<code>\$SystemResource[i].appliesTo</code>
conformsTo	Standard	UPDMElement	<code>\$SystemResource[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SystemResource[i].endBoundaryType</code>
milestone	ActualProjectMilestone	SystemResource	<code>\$SystemResource[i].milestone</code>
physicalLocation	ActualLocation	LocationHolder	<code>\$SystemResource[i].physicalLocation</code>
propertySet	PropertySet	UPDMElement	<code>\$SystemResource[i].propertySet</code>
requiredEnvironment	Environment	LocationHolder	<code>\$SystemResource[i].requiredEnvironment</code>
Resource.ownedOperation		SystemResource	<code>\$SystemResource[i].Resource.ownedOperation</code>
Resource.ownedPort		SystemResource	<code>\$SystemResource[i].Resource.ownedPort</code>
Resource.performs		SystemResource	<code>\$SystemResource[i].Resource.performs</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$SystemResource[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$SystemResource[i].URI</code>

994. Systems View MODAF

Base Classifier

- [InvisibleStereotype](#)

995. Systems Viewpoint**Base Classifier**

- [InvisibleStereotype](#)

996. systemValidationSuite**Base Classifier**

- [activeValidationSuite](#)

997. Table**Base Classifier**

- [Panel](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
tableColumns	Property	Table	<code>\$Table[i].tableColumns</code>
tableContextProperty	Property	Table	<code>\$Table[i].tableContextProperty</code>
tableContextPropertyTypes	Element	Table	<code>\$Table[i].tableContextPropertyTypes</code>
tableLayout	TableLayout	Table	<code>\$Table[i].tableLayout</code>
tableTitle	String	Table	<code>\$Table[i].tableTitle</code>

998. TableColumnName

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
tableColumn	Property	TableColumnName	<code>\$TableColumnName[i].tableColumn</code>
tableColumnName	String	TableColumnName	<code>\$TableColumnName[i].tableColumnName</code>

999. TableLayout

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Horizontal	Enumeration Literal	TableLayout	<code>\$TableLayout[i].Horizontal</code>
Vertical	Enumeration Literal	TableLayout	<code>\$TableLayout[i].Vertical</code>

1000. tagGroup

This stereotype should be applied to tag definition which values describes group names of other tag definitions. Used for tags grouping in MagicDraw model elements specifications, TaggedValues tabs.

1001. Technical Standards Viewpoint

Base Classifier

- [InvisibleStereotype](#)

1002. TechnicalStandard

MODAF: A ratified and peer-reviewed specification that is used to guide or constrain the architecture. A Standard may be applied to any element in the architecture via the [constrainedItem] property of UML::Constraint (Standard).

DoDAF: Technical standards document specific technical methodologies and practices to design and implement.

Base Classifier

- [Standard](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$TechnicalStandard[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$TechnicalStandard[i].conformsTo</code>
currentStatus	String	Standard	<code>\$TechnicalStandard[i].currentStatus</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$TechnicalStandard[i].endBoundaryType</code>
InformationTechnologyStandardCategory	String	Standard	<code>\$TechnicalStandard[i].InformationTechnologyStandardCategory</code>
mandatedDate	ISO8601DateTime	Standard	<code>\$TechnicalStandard[i].mandatedDate</code>
propertySet	PropertySet	UPDMElement	<code>\$TechnicalStandard[i].propertySet</code>
ratifiedBy	ActualOrganization	Standard	<code>\$TechnicalStandard[i].ratifiedBy</code>
retiredDate	ISO8601DateTime	Standard	<code>\$TechnicalStandard[i].retiredDate</code>
shortName	String	Standard	<code>\$TechnicalStandard[i].shortName</code>

startBoundaryType	ISO8601DateTime	UPDMElement	\$TechnicalStandard[i].startBoundaryType
URI	String	UPDMElement	\$TechnicalStandard[i].URI
version	String	Standard	\$TechnicalStandard[i].version

1003. TemporalPart

UPDM Artifact: An EnterprisePhase can be sub-divided into structural and temporal parts. TemporalPart describes the EnterprisePhase elements that have a time based nature.

MODAF: Asserts that one EnterprisePhase is a temporal part of another. Note: This means that both EnterprisePhases have the same spatial extent - i.e this is only a temporal structure (MODAF:: EnterpriseTemporalPart).

Base Classifier

- UPDMElement

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$TemporalPart[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$TemporalPart[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$TemporalPart[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$TemporalPart[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$TemporalPart[i].startBoundaryType
TemporalPart.class		TemporalPart	\$TemporalPart[i].TemporalPart.class
TemporalPart.type		TemporalPart	\$TemporalPart[i].TemporalPart.type
URI	String	UPDMElement	\$TemporalPart[i].URI

1004. TemporalScope

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
as is	Enumeration Literal	TemporalScope	\$TemporalScope[i].as is
current baseline	Enumeration Literal	TemporalScope	\$TemporalScope[i].current baseline
not applicable	Enumeration Literal	TemporalScope	\$TemporalScope[i].not applicable
not known	Enumeration Literal	TemporalScope	\$TemporalScope[i].not known
not specified	Enumeration Literal	TemporalScope	\$TemporalScope[i].not specified
planned baseline	Enumeration Literal	TemporalScope	\$TemporalScope[i].planned baseline
to be	Enumeration Literal	TemporalScope	\$TemporalScope[i].to be

1005. Term

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
synonyms	String	Term	<code>\$Term[i].synonyms</code>

1006. TestCase

A test case is a method for verifying a requirement is satisfied.

1007. TextDirection

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Horizontal	Enumeration Literal	TextDirection	<code>\$TextDirection[i].Horizontal</code>
Vertical	Enumeration Literal	TextDirection	<code>\$TextDirection[i].Vertical</code>

1008. TimeLine

1009. TimePeriod

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$TimePeriod[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$TimePeriod[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$TimePeriod[i].endBoundaryType</code>
endDate	ISO8601DateTime	TimePeriod	<code>\$TimePeriod[i].endDate</code>
propertySet	PropertySet	UPDMElement	<code>\$TimePeriod[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$TimePeriod[i].startBoundaryType</code>
startDate	ISO8601DateTime	TimePeriod	<code>\$TimePeriod[i].startDate</code>
URI	String	UPDMElement	<code>\$TimePeriod[i].URI</code>

1010. TODO_Owner

Base Classifier

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
TODO	String	TODO_Owner	\$TODO_Owner[i].TODO

1011. Trace

Specifies a trace relationship between model elements or sets of model elements that represent the same concept in different models. Traces are mainly used for tracking requirements and changes across models. Since model changes can occur in both directions, the directionality of the dependency can often be ignored. The mapping specifies the relationship between the two, but it is rarely computable and is usually informal.

1012. Trace

Base Classifier

- [DirectedRelationshipPropertyPath](#)
- [Trace](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getTracedFrom	Requirement	Trace	\$Trace[i].getTracedFrom
sourceContext	Classifier	DirectedRelationshipPropertyPath	\$Trace[i].sourceContext
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	\$Trace[i].sourcePropertyPath
targetContext	Classifier	DirectedRelationshipPropertyPath	\$Trace[i].targetContext
targetPropertyPath	Property	DirectedRelationshipPropertyPath	\$Trace[i].targetPropertyPath

1013. Transactional

A specialization of “InformationElement” the enables the specification of reusable information building blocks, upon which multiple community semantics can be built. Transactionals describe the constructions plans for data sets realizable from the underlying information/data store. The transactional links the community semantics to the structures and business rules information/data store.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
containedTransactionals	Transactional	Transactional	<code>\$Transactional[i].containedTransactionals</code>
identifier	Wrapper	Transactional	<code>\$Transactional[i].identifier</code>
representedWrappers	Wrapper	Transactional	<code>\$Transactional[i].representedWrappers</code>
Transactional.ownedAttribute		Transactional	<code>\$Transactional[i].Transactional.ownedAttribute</code>

1014. TransactionalAttribute

Specialization of Entity Attribute that enables the relationship between logical and Interim processing Attribute naming conventions.

1015. TransitionKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
external	Enumeration Literal	TransitionKind	<code>\$TransitionKind[i].external</code>
internal	Enumeration Literal	TransitionKind	<code>\$TransitionKind[i].internal</code>
local	Enumeration Literal	TransitionKind	<code>\$TransitionKind[i].local</code>

1016. treeStructureEnumeration

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
composition	Enumeration Literal	treeStructureEnumeration	<code>\$treeStructureEnumeration[i].composition</code>
containment	Enumeration Literal	treeStructureEnumeration	<code>\$treeStructureEnumeration[i].containment</code>
inheritance	Enumeration Literal	treeStructureEnumeration	<code>\$treeStructureEnumeration[i].inheritance</code>

1017. TriggerOnNestedPort

Base Classifier

- [ElementPropertyPath](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
onNestedPort	Port	TriggerOnNestedPort	<code>\$TriggerOnNestedPort[i].onNestedPort</code>
propertyPath	Property	ElementPropertyPath	<code>\$TriggerOnNestedPort[i].propertyPath</code>

1018. Trustline

MODAF: Asserts that the trustingParty (either a Node or a KnownResource) trusts the trustedParty to a given level (indicated by the level attribute). Note: No unit of measure is associated with the level - security architects must define their own scale of trust levels for a given architecture or set of architectures.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Trustline[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Trustline[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Trustline[i].endBoundaryType</code>
level	String	Trustline	<code>\$Trustline[i].level</code>
propertySet	PropertySet	UPDMElement	<code>\$Trustline[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Trustline[i].startBoundaryType</code>
Trustline.client		Trustline	<code>\$Trustline[i].Trustline.client</code>
Trustline.supplier		Trustline	<code>\$Trustline[i].Trustline.supplier</code>
URI	String	UPDMElement	<code>\$Trustline[i].URI</code>

1019. TV-1

Standard Profile (TV-1) table defines the technical and non technical standards, guidance and policy applicable to the architecture.

There are two ways to add a row in this table:

1. Add new UPDM Element. Click “**Add new UPDM Element**” button. Select UPDM Element (this term means any of the UPDM available element). Specify the owner for the selected element.

2. Add Existing UPDM elements. Click “**Add Existing UPDM Element**” button and select UPDM elements.

For each row there should be assigned one or more Standards or Protocols that the particular UPDM element must conform to. Click “...” button on any cell in the "Standard/Policy" column.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

1020. TV-1 Package

Base Classifier

- [InvisibleStereotype](#)

1021. TV-2

Standards Forecast (SV-2) table defines expected changes in technology related standards and conventions.

You will find this table identical to the SV-9 table. In general they are identical in implementation, but TV-2 is more likely to be used for Standards and Protocols forecasting.

Three major steps should be done to create the table:

1. Add Rows to the Table. There are two ways to add a row to the table:

*1.1. Add new Subject of Forecast as row Header. Click **Add New** button and select element you want to create. Specify owner for selected element.*

*1.2. Add Existing Subject of Forecast as row Header. Click **Add Existing** button and select one or more existing elements.*

*2. Add columns to the table. Click "**Add/Remove forecast**" button. Specify Time Periods for the forecasting: select or create Time Line Package to store Forecast dates; select forecast kind. According to the selected Forecast Kind, specify additional options needed (see Forecasting Period Dialog for more info).*

3. Fill in the cells with the Subjects of Forecast. Click the "..." button on the cell you want to fill in, select Resources, Standards, Protocols or Competences to forecast. Specify a date for the Forecast. By default this date will be the starting date of the selected Time Period.

Rows can be removed from model or only from table, can be ordered and exported to the CSV or HTML.

More Actions are available by clicking right mouse button on a cell.

Base Classifier

- [InvisibleStereotype](#)

1022. TV-2 Package

Base Classifier

- [InvisibleStereotype](#)

1023. Type

A class that specifies a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those objects. However, it may have attributes and associations. Behavioral specifications for type operations may be expressed using, for example, activity diagrams. An object may have at most one implementation class, however it may conform to multiple different types. See also: «implementationClass».

1024. typeModifier**Base Classifier**

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
typeModifier	String	typeModifier	<code>\$typeModifier[i].typeModifier</code>

1025. typeModifierEnumeration

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
&	Enumeration Literal	typeModifierEnumeration	<code>\$typeModifierEnumeration[i].&</code>
*	Enumeration Literal	typeModifierEnumeration	<code>\$typeModifierEnumeration[i].*</code>
[]	Enumeration Literal	typeModifierEnumeration	<code>\$typeModifierEnumeration[i].[]</code>

1026. Uniform

Uniform distribution - constant probability between min and max

Base Classifier

- [BasicInterval](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
max	Real	BasicInterval	\$Uniform[i].max
min	Real	BasicInterval	\$Uniform[i].min

1027. Unit

A Unit is a quantity in terms of which the magnitudes of other quantities that have the same dimension can be stated. A unit often relies on precise and reproducible ways to measure the unit. For example, a unit of length such as meter may be specified as a multiple of a particular wavelength of light. A unit may also specify less stable or precise ways to express some value, such as a cost expressed in some currency, or a severity rating measured by a numerical scale.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
definitionURI	String	Unit	\$Unit[i].definitionURI
description	String	Unit	\$Unit[i].description
quantityKind	QuantityKind	Unit	\$Unit[i].quantityKind
symbol	String	Unit	\$Unit[i].symbol

1028. UnlimitedNatural

Base Classifier

- Number

1029. UPDMElement

UPDM Artifact: Super type for many of the UPDM elements. It provides a means of extending UPDM elements in a common way. With links to the measurement set, it also allows quantitative metrics to be associated with structural and behavioral elements.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$UPDMElement[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$UPDMElement[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$UPDMElement[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$UPDMElement[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$UPDMElement[i].startBoundaryType
URI	String	UPDMElement	\$UPDMElement[i].URI

1030. usabilityRequirement

Requirement about usability.

Base Classifier

- [extendedRequirement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Derived	Requirement	Requirement	<code>\$usabilityRequirement[i].Derived</code>
DerivedFrom	Requirement	Requirement	<code>\$usabilityRequirement[i].DerivedFrom</code>
Id	String	Requirement	<code>\$usabilityRequirement[i].Id</code>
Master	Requirement	Requirement	<code>\$usabilityRequirement[i].Master</code>
RefinedBy	NamedElement	Requirement	<code>\$usabilityRequirement[i].RefinedBy</code>
risk	RiskKind	extendedRequirement	<code>\$usabilityRequirement[i].risk</code>
SatisfiedBy	NamedElement	Requirement	<code>\$usabilityRequirement[i].SatisfiedBy</code>
source	String	extendedRequirement	<code>\$usabilityRequirement[i].source</code>
Text	String	Requirement	<code>\$usabilityRequirement[i].Text</code>
TracedTo	NamedElement	Requirement	<code>\$usabilityRequirement[i].TracedTo</code>
VerifiedBy	NamedElement	Requirement	<code>\$usabilityRequirement[i].VerifiedBy</code>
verifyMethod	VerificationMethodKind	extendedRequirement	<code>\$usabilityRequirement[i].verifyMethod</code>

1031. useCaseModel

A use case model specifies the services a system provides to its users; that is, the different ways of using the system, and whose top-level package is a use case system.

1032. useCaseView**Base Classifier**

- [InvisibleStereotype](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
useCaseViewID	int	useCaseView	<code>\$useCaseView[i].useCaseViewID</code>

1033. User system

An User system is a special external system that serves as medium between a user and the system without having own interests in the communication. For example Input device or Display.

Base Classifier

- [External system](#)

1034. Utilities

Base Classifier

- [InvisibleStereotype](#)

1035. Utility

A class that has no instances, but rather denotes a named collection of non-member attributes and operations, all of which are class-scoped.

1036. validationRule

Base Classifier

- [invariant](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
abbreviation	String	validationRule	\$validationRule[i].abbreviation
errorMessage	String	validationRule	\$validationRule[i].errorMessage
implementation	String	validationRule	\$validationRule[i].implementation
parentObject	StructuredExpression	validationRule	\$validationRule[i].parentObject
parentRule	validationRule	validationRule	\$validationRule[i].parentRule
severity	SeverityKind	validationRule	\$validationRule[i].severity

1037. validationSuite

1038. ValueProperty

A value property specifies a quantitative property of its containing block. Every value property is typed by either a SysML value type or a UML data type. A value property will be displayed under the 'values' compartment when 'Sort by SysML Style' in the 'Presentation Options' is selected.

Base Classifier

- [InvisibleStereotype](#)

1039. ValueType

A *ValueType* defines types of values that may be used to express information about a system, but cannot be identified as the target of any reference. Since a value cannot be identified except by means of the value itself, each such value within a model is independent of any other, unless other forms of constraints are imposed. Value types may be used to type properties, operation parameters, or potentially other elements within SysML. SysML defines *ValueType* as a stereotype of UML *DataType* to establish a more neutral term for system values that may never be given a concrete data representation.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
quantityKind	InstanceSpecification	ValueType	\$ValueType[i].quantityKind
unit	InstanceSpecification	ValueType	\$ValueType[i].unit

1040. Variable

1041. Variant Map

The **Variant Map** depicts specializations of a **System Resource** including Service Access, Capability Configuration, Software and other elements. Specializations of a System Resource are modeled using the Generalization relationship.

The model structure in the Relation Map diagram can be discovered in two different **layouts**: **tree** or **radial**. The relations represented in the diagram can be analyzed in the following ways:



-Dynamically - according to the applied filters: element type, dependencies criterion, depth, and scope.

-Statically - step by step expanding the branches of the structure.

Manipulation:


Create a relation map structure - drag and drop element from the Containment Tree on the Relation Map.

Restore manually suppressed / expanded branches and hidden elements and they position - click Restore Layout button .

Expand / suppress branches - click on smart manipulator after the Node  / .

Move the whole structure - click on the empty place in the Relation Map and drag.


Move the selected Node - click on the Node and drag.

Insert New Element - select an element and click  to choose an element and relationship. Press Insert (Cmd+I for MAC users) to create an element faster and more easily.

Insert New Element of the Same Type - type the name and press Ctrl+Enter to create an element of the same type.

Zoom in - Ctrl + mouse wheel scroll up.

Zoom out - Ctrl + mouse wheel scroll down.

Change the Context element on demand by selecting a new Node – on the Relation Map toolbar click **Make element as context on selection** button , select Node which should be set as context.

Change the Context element – select another node and click **Set element as context** on its shortcut menu or on the Relation Map toolbar .

Base Classifier

- InvisibleStereotype

1042. VerdictKind

Type of a return parameter of a TestCase must be VerdictKind, consistent with the UML Testing Profile.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
error	Enumeration Literal	VerdictKind	\$VerdictKind[i].error
fail	Enumeration Literal	VerdictKind	\$VerdictKind[i].fail
inconclusive	Enumeration Literal	VerdictKind	\$VerdictKind[i].inconclusive
pass	Enumeration Literal	VerdictKind	\$VerdictKind[i].pass

1043. VerificationMethodKind

- 1) Analysis indicates that verification will be performed by technical evaluation using mathematical representations, charts, graphs, circuit diagrams, data reduction, or representative data. Analysis also includes the verification of requirements under conditions, which are simulated or modeled; where the results are derived from the analysis of the results produced by the model,
- 2) Demonstration indicates that verification will be performed by operation, movement or adjustment of the item under specific conditions to perform the design functions without recording of quantitative data. Demonstration is typically considered the least restrictive of the verification types,
- 3) Inspection indicates that verification will be performed by examination of the item, reviewing descriptive documentation, and comparing the appropriate characteristics with a predetermined standard to determine conformance to requirements without the use of special laboratory equipment or procedures, and
- 4) Test indicates that verification will be performed through systematic exercising of the applicable item under appropriate conditions with instrumentation to measure required parameters and the collection, analysis, and evaluation of quantitative data to show that measured parameters equal or exceed specified requirements.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Analysis	Enumeration Literal	VerificationMethodKind	\$VerificationMethodKind[i].Analysis
Demonstration	Enumeration Literal	VerificationMethodKind	\$VerificationMethodKind[i].Demonstration
Inspection	Enumeration Literal	VerificationMethodKind	\$VerificationMethodKind[i].Inspection
Test	Enumeration Literal	VerificationMethodKind	\$VerificationMethodKind[i].Test

1044. Verify

A *Verify* relationship is a dependency between a requirement and a test case or other model element that can determine whether a system fulfills the requirement. As with other dependencies, the arrow direction points from the (client) element to the (supplier) requirement.

Base Classifier

- Trace

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
getTracedFrom	Requirement	Trace	\$Verify[i].getTracedFrom
getVerifies	Requirement	Verify	\$Verify[i].getVerifies
sourceContext	Classifier	DirectedRelationshipPropertyPath	\$Verify[i].sourceContext
sourcePropertyPath	Property	DirectedRelationshipPropertyPath	\$Verify[i].sourcePropertyPath
targetContext	Classifier	DirectedRelationshipPropertyPath	\$Verify[i].targetContext
targetPropertyPath	Property	DirectedRelationshipPropertyPath	\$Verify[i].targetPropertyPath

1045. VersionOfConfiguration

MODAF: Asserts that a *CapabilityConfiguration* is a version of a *WholeLifeConfiguration*.

DoDAF: NA

Base Classifier

- UPDMElement

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$VersionOfConfiguration[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$VersionOfConfiguration[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$VersionOfConfiguration[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$VersionOfConfiguration[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$VersionOfConfiguration[i].startBoundaryType
URI	String	UPDMElement	\$VersionOfConfiguration[i].URI
VersionOfConfiguration.class		VersionOfConfiguration	\$VersionOfConfiguration[i].VersionOfConfiguration.class
VersionOfConfiguration.type		VersionOfConfiguration	\$VersionOfConfiguration[i].VersionOfConfiguration.type

1046. View

MODAF: A specification of a way to present an aspect of the architecture. Views are defined with one or more purposes in mind - e.g. showing the logical topology of the enterprise, describing a process model, defining a data model, etc.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$View[i].actualPropertySet</code>
architecturalElements	UPDMElement	View	<code>\$View[i].architecturalElements</code>
conformsTo	Standard	UPDMElement	<code>\$View[i].conformsTo</code>
coversPhase	EnterprisePhase	View	<code>\$View[i].coversPhase</code>
description	String	View	<code>\$View[i].description</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$View[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$View[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$View[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$View[i].URI</code>
viewpoints	Viewpoint	View	<code>\$View[i].viewpoints</code>

1047. View

A View is a representation of a whole system or subsystem from the perspective of a single viewpoint. Views are allowed to import other elements including other packages and other views that conform to the viewpoint.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
stakeholder	Stakeholder	View	<code>\$View[i].stakeholder</code>
viewPoint	Viewpoint	View	<code>\$View[i].viewPoint</code>

1048. View**Base Classifier**

- [InvisibleStereotype](#)
- [View](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
ID	String	View	<code>\$View[i].ID</code>
stakeholder	Stakeholder	View	<code>\$View[i].stakeholder</code>
viewPoint	Viewpoint	View	<code>\$View[i].viewPoint</code>

1049. Viewpoint

MODAF: An instance of the specified View.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Viewpoint[i].actualPropertySet</code>
concerns	String	Viewpoint	<code>\$Viewpoint[i].concerns</code>
conformsTo	Standard	UPDMElement	<code>\$Viewpoint[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Viewpoint[i].endBoundaryType</code>
languages	String	Viewpoint	<code>\$Viewpoint[i].languages</code>
methods	String	Viewpoint	<code>\$Viewpoint[i].methods</code>
propertySet	PropertySet	UPDMElement	<code>\$Viewpoint[i].propertySet</code>
purpose	String	Viewpoint	<code>\$Viewpoint[i].purpose</code>
stakeholders	String	Viewpoint	<code>\$Viewpoint[i].stakeholders</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Viewpoint[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$Viewpoint[i].URI</code>

1050. Viewpoint

A Viewpoint is a specification of the conventions and rules for constructing and using a view for the purpose of addressing a set of stakeholder concerns. The languages and methods for specifying a view may reference languages and methods in another viewpoint. They specify the elements expected to be represented in the view, and may be formally or informally defined. For example, the security viewpoint may require the security requirements, security functional and physical architecture, and security test cases.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
concern	String	Viewpoint	<code>\$Viewpoint[i].concern</code>
concernList	Comment	Viewpoint	<code>\$Viewpoint[i].concernList</code>
language	String	Viewpoint	<code>\$Viewpoint[i].language</code>
method	Behavior	Viewpoint	<code>\$Viewpoint[i].method</code>
presentation	String	Viewpoint	<code>\$Viewpoint[i].presentation</code>
purpose	String	Viewpoint	<code>\$Viewpoint[i].purpose</code>
stakeholder	Stakeholder	Viewpoint	<code>\$Viewpoint[i].stakeholder</code>

1051. virtual

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
details	String	virtual	<code>\$virtual[i].details</code>
n	int	virtual	<code>\$virtual[i].n</code>

1052. VisibilityKind

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
package	Enumeration Literal	VisibilityKind	<code>\$VisibilityKind[i].package</code>
private	Enumeration Literal	VisibilityKind	<code>\$VisibilityKind[i].private</code>
protected	Enumeration Literal	VisibilityKind	<code>\$VisibilityKind[i].protected</code>
public	Enumeration Literal	VisibilityKind	<code>\$VisibilityKind[i].public</code>

1053. Vision

MODAF: The overall aims of an enterprise over a given period of time. (EnterpriseVision)

DoDAF: An end that describes the future state of the enterprise, without regard to how it is to be achieved; a mental image of what the future will or could be like.

Base Classifier

- [EnterpriseVision](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$Vision[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$Vision[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Vision[i].endBoundaryType</code>
enterprisePhase	EnterprisePhase	EnterpriseVision	<code>\$Vision[i].enterprisePhase</code>
propertySet	PropertySet	UPDMElement	<code>\$Vision[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$Vision[i].startBoundaryType</code>
statement	VisionStatement	EnterpriseVision	<code>\$Vision[i].statement</code>
URI	String	UPDMElement	<code>\$Vision[i].URI</code>

1054. VisionStatement

MODAF: A high-level textual description of an EnterpriseVision.

DoDAF: An end that describes the future state of the enterprise, without regard to how it is to be achieved; a mental image of what the future will or could be like (DODAF::Vision).

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$VisionStatement[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$VisionStatement[i].conformsTo</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$VisionStatement[i].endBoundaryType</code>
propertySet	PropertySet	UPDMElement	<code>\$VisionStatement[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$VisionStatement[i].startBoundaryType</code>
URI	String	UPDMElement	<code>\$VisionStatement[i].URI</code>

1055. warningIcon**Base Classifier**

- [imaged](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
highlightColor	String	imaged	<code>\$warningIcon[i].highlightColor</code>

1056. WebReportNodeName

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
reportCategory	ReportCategory	WebReportNodeName	<code>\$WebReportNodeName[i].reportCategory</code>



1057. WhiteBoxICDTable

A Whitebox ICD Table represents assembly – how Parts are connected via Ports and interfaces.

With this table you can easily:

- Review Connectors of the Parts in the single place.
- Customize the representation of the table.
- Export the data into an *.html, *.csv, or *.xlsx file.

Toolbar button descriptions:

- **Delete** – click to remove selected elements both from the table and from the model.
- **Remove From Table** - click to remove selected elements from the table.
- **Refresh** – click  to update the contents of the table after specifying the Block as the context.
- **Validate Diagram** – click  to validate the diagram.
- **Up** – click to shift selected elements (either grouped or non-grouped) up a row.
- **Down** – click to shift selected elements (either grouped or non-grouped) down a row.
- **Show Columns** – click to specify the set of columns for displaying in the table.
- **Show Parts** – click to specify the set of Parts for displaying Connectors between them in the table.
- **Options** > **Show Full Paths** – click to display the full paths of elements in the table.
- **Export** - click to export the contents of the table to an *.html, *.csv, or *.xlsx file.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
query	Element	WhiteBoxICDTable	\$WhiteBoxICDTable[i].query

1058. WholeLifeConfiguration

MODAF: A set of versions of a CapabilityConfiguration over time.

DoDAF: NA

Base Classifier

- [UPDMElement](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	\$WholeLifeConfiguration[i].actualPropertySet
conformsTo	Standard	UPDMElement	\$WholeLifeConfiguration[i].conformsTo
endBoundaryType	ISO8601DateTime	UPDMElement	\$WholeLifeConfiguration[i].endBoundaryType
propertySet	PropertySet	UPDMElement	\$WholeLifeConfiguration[i].propertySet
startBoundaryType	ISO8601DateTime	UPDMElement	\$WholeLifeConfiguration[i].startBoundaryType
URI	String	UPDMElement	\$WholeLifeConfiguration[i].URI

1059. WholeLifeEnterprise

UPDM: A WholeLifeEnterprise is a purposeful endeavor of any size involving people, organizations and supporting systems (including physical systems and/or processes).

MODAF: An EnterprisePhase that represents the whole existence of an enterprise.

DoDAF: NA

Base Classifier

- [EnterprisePhase](#)

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
actualPropertySet	ActualPropertySet	UPDMElement	<code>\$WholeLifeEnterprise[i].actualPropertySet</code>
conformsTo	Standard	UPDMElement	<code>\$WholeLifeEnterprise[i].conformsTo</code>
describedBy	ArchitecturalDescription	EnterprisePhase	<code>\$WholeLifeEnterprise[i].describedBy</code>
endBoundaryType	ISO8601DateTime	UPDMElement	<code>\$WholeLifeEnterprise[i].endBoundaryType</code>
endDate	ISO8601DateTime	EnterprisePhase	<code>\$WholeLifeEnterprise[i].endDate</code>
Enterprise from/to		EnterprisePhase	<code>\$WholeLifeEnterprise[i].Enterprise from/to</code>
EnterprisePhase.useCase		EnterprisePhase	<code>\$WholeLifeEnterprise[i].EnterprisePhase.useCase</code>
fulfills	Mission	EnterprisePhase	<code>\$WholeLifeEnterprise[i].fulfills</code>
goals	EnterpriseGoal	EnterprisePhase	<code>\$WholeLifeEnterprise[i].goals</code>
propertySet	PropertySet	UPDMElement	<code>\$WholeLifeEnterprise[i].propertySet</code>
startBoundaryType	ISO8601DateTime	UPDMElement	<code>\$WholeLifeEnterprise[i].startBoundaryType</code>
startDate	ISO8601DateTime	EnterprisePhase	<code>\$WholeLifeEnterprise[i].startDate</code>
statementTasks	EnduringTask	EnterprisePhase	<code>\$WholeLifeEnterprise[i].statementTasks</code>
URI	String	UPDMElement	<code>\$WholeLifeEnterprise[i].URI</code>
visions	EnterpriseVision	EnterprisePhase	<code>\$WholeLifeEnterprise[i].visions</code>

1060. Wrapper

A specialization of “EntityItem” that links a Transactional to the logical information/data model Elements (e.g., DB Table). Wrappers represent a single instance of “EntityItem” data.

Attribute Name	Attribute Type	Attribute Owner	Sample Template Expression (VTL)
Wrapper.ownedAttribute		Wrapper	<code>\$Wrapper[i].Wrapper.ownedAttribute</code>

1061. WrapperAttribute

Specialization of Entity Attribute that enables the relationship between physical and logical attribute naming conventions.