OntoDiff: lex\_sp-defs-231113 vs lex\_sp-defs-231012

==== === === [ OntoRail Diff ] === === ====
 • target: lex\_sp-defs-231113 (https://glossaries.ontorail.org/LEX\_SP-DEFS/lex\_sp-defs-231113#)
 • versus: lex\_sp-defs-231012 (https://glossaries.ontorail.org/LEX\_SP-DEFS/lex\_sp-defs-231012#)
 • entity types considered: ['lexinfo:AbbreviatedForm', 'ontolex:LexicalEntry', 'ontolex:Form', 'ontolex:LexicalSense', 'ontolex:LexicalConcept']
 • performed: 2024-01-11 16:42:51 +0000
 • duration: 4.5 sec
 • OntoDiff version date: 2024-01-11 16:37:49
 • Ignored predicates: xmi:ea\_localid, xmi:lowerValue\_\_id, xmi:upperValue\_\_id, xmi:source\_\_isNavigable, xmi:coords\_\_ordered, xmi:coords\_\_scale, xmi:containment\_\_position, xmi:virtualInheritance, xmi:target\_\_isNavigable, xmi:source\_\_idref, xmi:target\_\_idref, xmi:type\_\_idref, xmi:labels\_\_rb, xmi:type, xmi:visibility, xmi:isUnique, xmi:upperValue\_\_type, xmi:isDerived, xmi:isDerivedUnion, xmi:isOrdered, xmi:isReadOnly, xmi:isStatic
=== === === === === === === === === === ===

Table Of Content

|  |  |
| --- | --- |
| Modifications Summary | Modifications Details |
|  \* [lexinfo:AbbreviatedForm](#Summary_lexinfo:AbbreviatedForm) \* [ontolex:LexicalEntry](#Summary_ontolex:LexicalEntry) \* [ontolex:Form](#Summary_ontolex:Form) \* [ontolex:LexicalSense](#Summary_ontolex:LexicalSense) \* [ontolex:LexicalConcept](#Summary_ontolex:LexicalConcept) |  \* [lexinfo:AbbreviatedForm](#Details_lexinfo:AbbreviatedForm) \* [ontolex:LexicalEntry](#Details_ontolex:LexicalEntry) \* [ontolex:Form](#Details_ontolex:Form) \* [ontolex:LexicalSense](#Details_ontolex:LexicalSense) \* [ontolex:LexicalConcept](#Details_ontolex:LexicalConcept) |

# Summary

## lexinfo:AbbreviatedForm entities

### 12 lexinfo:AbbreviatedForm in lex\_sp-defs-231113:

### 5 lexinfo:AbbreviatedForm NEW from lex\_sp-defs-231012:

CCS, ERJU, EU-Rail, RAMS, TCCS

### 0 lexinfo:AbbreviatedForm REMOVED from lex\_sp-defs-231012:

### 0 lexinfo:AbbreviatedForm MODIFIED from lex\_sp-defs-231012:

## ontolex:LexicalEntry entities

### 661 ontolex:LexicalEntry in lex\_sp-defs-231113:

### 56 ontolex:LexicalEntry NEW from lex\_sp-defs-231012:

"AB Allocation Body", "Application Guideline", CCS, "CI Common Interface", "CMS Capacity Management System", "Common standard properties of workitems", "Configuration Data", "Control, Command and Signalling", "DCM Digital Capacity Management", Def, "ECMT European Capacity Management Tool", EEE, "ERG - Euro Radio Gateway", ERJU, "ERTMS comprises of the European Train Control System (ETCS), i.e. a cab-signalling system that incorporates automatic train protection, the Global System for Mobile communications for Railways (GSM-R) and operating rules.", "ETA Estimated Time of Arrival", EU-Rail, "Engineering Input Data", "Essential Function", "Europe's Rail", "GUI Graphical User Interface", "HMI Component", IM, "Infrastructure Manager", "Mission Assignment", "Network Component", "Operational Mission", "Operational Mission Segment", "Operational Stop", "PCS Path Coordination System", RAMS, "RIM Railway Infrastructure Manager", "RIS Railway Infrastructure System", "ROC Railway Operating Company", Railway, "Reliability, Availability, Maintainability and Safety", "Reliable Data", "Road Traffic (DRAFT)", "Roll Away", "SR Authorisation", "Secure Communication", "Secure Component", "Shared Security Services", "TAF/TAP TSI Technical Specification for Interoperability relating to Telematics...", TCCS, "TCR Temporary Capacity Restriction", "TIMS - Train Integrity Management System", "TMS Traffic Management System", "Train Running Number", "Transversal CCS domain", "Wireless Component", "empty definition", "fault correction time", "fault localization time", "function checkout time", new

### 9 ontolex:LexicalEntry REMOVED from lex\_sp-defs-231012:

"AB CI CMS DCM ECMT GUI ETA PCS RIM RIS ROC TAF/TAP TSI TCR TMS Allocation Body C...", "Common starndard properties of workitems", FCKT, FCT, FLT, "IM - Infrastructure Manager", Mitigation, "TIMS - Train Integrity", "Targetsystem: a subsystem of the System Pillar’s architecture which is a target..."

### 0 ontolex:LexicalEntry MODIFIED from lex\_sp-defs-231012:

## ontolex:Form entities

### 661 ontolex:Form in lex\_sp-defs-231113:

### 56 ontolex:Form NEW from lex\_sp-defs-231012:

AB--ALLOCATION--BODY\_lexForm, APPLICATION--GUIDELINE\_lexForm, CI--COMMON--INTERFACE\_lexForm, CMS--CAPACITY--MANAGEMENT--SYSTEM\_lexForm, COMMON--STANDARD--PROPERTIES--OF--WORKITEMS\_lexForm, CONFIGURATION--DATA\_lexForm, CONTROL--COMMAND--AND--SIGNALLING\_lexForm, CONTROL--COMMAND--AND--SIGNALLING\_lexForm\_2, DCM--DIGITAL--CAPACITY--MANAGEMENT\_lexForm, DEF\_lexForm, ECMT--EUROPEAN--CAPACITY--MANAGEMENT--TOOL\_lexForm, EEE\_lexForm, EMPTY--DEFINITION\_lexForm, ENGINEERING--INPUT--DATA\_lexForm, ERG-----EURO--RADIO--GATEWAY\_lexForm, ERJU\_lexForm\_2, ERTMS--COMPRISES--OF--THE--EUROPEAN--TRAIN--CONTROL--SYSTEM--ETCS--I-E---A--CAB-SIGNALLING--SYSTEM--THAT--INCORPORATES--AUTOMATIC--TRAIN--PROTECTION--THE--GLOBAL--SYSTEM--FOR--MOBILE--COMMUNICATIONS--FOR--RAILWAYS--GSM-R--AND--OPERATING--RULES\_lexForm, ESSENTIAL--FUNCTION\_lexForm, ETA--ESTIMATED--TIME--OF--ARRIVAL\_lexForm, EUROPE\_S--RAIL\_lexForm, EUROPE\_S--RAIL\_lexForm\_2, FAULT--CORRECTION--TIME\_lexForm, FAULT--LOCALIZATION--TIME\_lexForm, FUNCTION--CHECKOUT--TIME\_lexForm, GUI--GRAPHICAL--USER--INTERFACE\_lexForm, HMI--COMPONENT\_lexForm, INFRASTRUCTURE--MANAGER\_lexForm, INFRASTRUCTURE--MANAGER\_lexForm\_2, MISSION--ASSIGNMENT\_lexForm, NETWORK--COMPONENT\_lexForm, NEW\_lexForm, OPERATIONAL--MISSION--SEGMENT\_lexForm, OPERATIONAL--MISSION\_lexForm, OPERATIONAL--STOP\_lexForm, PCS--PATH--COORDINATION--SYSTEM\_lexForm, RAILWAY\_lexForm, RELIABILITY--AVAILABILITY--MAINTAINABILITY--AND--SAFETY\_lexForm, RELIABILITY--AVAILABILITY--MAINTAINABILITY--AND--SAFETY\_lexForm\_2, RELIABLE--DATA\_lexForm, RIM--RAILWAY--INFRASTRUCTURE--MANAGER\_lexForm, RIS--RAILWAY--INFRASTRUCTURE--SYSTEM\_lexForm, ROAD--TRAFFIC--DRAFT\_lexForm, ROC--RAILWAY--OPERATING--COMPANY\_lexForm, ROLL--AWAY\_lexForm, SECURE--COMMUNICATION\_lexForm, SECURE--COMPONENT\_lexForm, SHARED--SECURITY--SERVICES\_lexForm, SR--AUTHORISATION\_lexForm, TAF\_TAP--TSI--TECHNICAL--SPECIFICATION--FOR--INTEROPERABILITY--RELATING--TO--TELEMATICS\_lexForm, TCR--TEMPORARY--CAPACITY--RESTRICTION\_lexForm, TIMS-----TRAIN--INTEGRITY--MANAGEMENT--SYSTEM\_lexForm, TMS--TRAFFIC--MANAGEMENT--SYSTEM\_lexForm, TRAIN--RUNNING--NUMBER\_lexForm, TRANSVERSAL--CCS--DOMAIN\_lexForm, TRANSVERSAL--CCS--DOMAIN\_lexForm\_2, WIRELESS--COMPONENT\_lexForm

### 9 ontolex:Form REMOVED from lex\_sp-defs-231012:

AB--CI--CMS--DCM--ECMT--GUI--ETA--PCS--RIM--RIS--ROC--TAF\_TAP--TSI--TCR--TMS--ALLOCATION--BODY--C\_lexForm, COMMON--STARNDARD--PROPERTIES--OF--WORKITEMS\_lexForm, FCKT\_lexForm, FCT\_lexForm, FLT\_lexForm, IM-----INFRASTRUCTURE--MANAGER\_lexForm, MITIGATION\_lexForm, TARGETSYSTEM---A--SUBSYSTEM--OF--THE--SYSTEM--PILLAR\_S--ARCHITECTURE--WHICH--IS--A--TARGET\_lexForm, TIMS-----TRAIN--INTEGRITY\_lexForm

### 0 ontolex:Form MODIFIED from lex\_sp-defs-231012:

## ontolex:LexicalSense entities

### 635 ontolex:LexicalSense in lex\_sp-defs-231113:

### 50 ontolex:LexicalSense NEW from lex\_sp-defs-231012:

AB--ALLOCATION--BODY\_lexSense, APPLICATION--GUIDELINE\_lexSense, CI--COMMON--INTERFACE\_lexSense, CMS--CAPACITY--MANAGEMENT--SYSTEM\_lexSense, COMMON--STANDARD--PROPERTIES--OF--WORKITEMS\_lexSense, CONFIGURATION--DATA\_lexSense, CONTROL--COMMAND--AND--SIGNALLING\_lexSense, DCM--DIGITAL--CAPACITY--MANAGEMENT\_lexSense, DEF\_lexSense, ECMT--EUROPEAN--CAPACITY--MANAGEMENT--TOOL\_lexSense, EEE\_lexSense, EMPTY--DEFINITION\_lexSense, ENGINEERING--INPUT--DATA\_lexSense, ERG-----EURO--RADIO--GATEWAY\_lexSense, ERTMS--COMPRISES--OF--THE--EUROPEAN--TRAIN--CONTROL--SYSTEM--ETCS--I-E---A--CAB-SIGNALLING--SYSTEM--THAT--INCORPORATES--AUTOMATIC--TRAIN--PROTECTION--THE--GLOBAL--SYSTEM--FOR--MOBILE--COMMUNICATIONS--FOR--RAILWAYS--GSM-R--AND--OPERATING--RULES\_lexSense, ESSENTIAL--FUNCTION\_lexSense, ETA--ESTIMATED--TIME--OF--ARRIVAL\_lexSense, EUROPE\_S--RAIL\_lexSense, FAULT--CORRECTION--TIME\_lexSense, FAULT--LOCALIZATION--TIME\_lexSense, FUNCTION--CHECKOUT--TIME\_lexSense, GUI--GRAPHICAL--USER--INTERFACE\_lexSense, HMI--COMPONENT\_lexSense, INFRASTRUCTURE--MANAGER\_lexSense, MISSION--ASSIGNMENT\_lexSense, NETWORK--COMPONENT\_lexSense, NEW\_lexSense, OPERATIONAL--MISSION--SEGMENT\_lexSense, OPERATIONAL--MISSION\_lexSense, OPERATIONAL--STOP\_lexSense, PCS--PATH--COORDINATION--SYSTEM\_lexSense, RAILWAY\_lexSense, RELIABILITY--AVAILABILITY--MAINTAINABILITY--AND--SAFETY\_lexSense, RELIABLE--DATA\_lexSense, RIM--RAILWAY--INFRASTRUCTURE--MANAGER\_lexSense, RIS--RAILWAY--INFRASTRUCTURE--SYSTEM\_lexSense, ROAD--TRAFFIC--DRAFT\_lexSense, ROC--RAILWAY--OPERATING--COMPANY\_lexSense, ROLL--AWAY\_lexSense, SECURE--COMMUNICATION\_lexSense, SECURE--COMPONENT\_lexSense, SHARED--SECURITY--SERVICES\_lexSense, SR--AUTHORISATION\_lexSense, TAF\_TAP--TSI--TECHNICAL--SPECIFICATION--FOR--INTEROPERABILITY--RELATING--TO--TELEMATICS\_lexSense, TCR--TEMPORARY--CAPACITY--RESTRICTION\_lexSense, TIMS-----TRAIN--INTEGRITY--MANAGEMENT--SYSTEM\_lexSense, TMS--TRAFFIC--MANAGEMENT--SYSTEM\_lexSense, TRAIN--RUNNING--NUMBER\_lexSense, TRANSVERSAL--CCS--DOMAIN\_lexSense, WIRELESS--COMPONENT\_lexSense

### 9 ontolex:LexicalSense REMOVED from lex\_sp-defs-231012:

AB--CI--CMS--DCM--ECMT--GUI--ETA--PCS--RIM--RIS--ROC--TAF\_TAP--TSI--TCR--TMS--ALLOCATION--BODY--C\_lexSense, COMMON--STARNDARD--PROPERTIES--OF--WORKITEMS\_lexSense, FCKT\_lexSense, FCT\_lexSense, FLT\_lexSense, IM-----INFRASTRUCTURE--MANAGER\_lexSense, MITIGATION\_lexSense, TARGETSYSTEM---A--SUBSYSTEM--OF--THE--SYSTEM--PILLAR\_S--ARCHITECTURE--WHICH--IS--A--TARGET\_lexSense, TIMS-----TRAIN--INTEGRITY\_lexSense

### 40 ontolex:LexicalSense MODIFIED from lex\_sp-defs-231012:

ADAPTABILITY\_lexSense, AUTOMATIC--CONFLICT--SOLUTION\_lexSense, CAPABILITY--MEANING\_lexSense, CAPACITY--PLAN--AND--DECISION--PROCESSING\_lexSense, CAPACITY--WASTE--MEANS\_lexSense, CAPACITY-FRIENDLY--BEHAVIOUR--MEANS\_lexSense, CAPEX\_lexSense, CHANGEABILITY\_lexSense, DEVIATION--DETECTION\_lexSense, ERTMS\_lexSense, EVOLVABILITY\_lexSense, EXCHANGEABILITY\_lexSense, FFFIS-----FORM--FIT--FUNCTIONAL--INTERFACE--SPECIFICATION\_lexSense, FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexSense, FUNCTIONAL--APPORTIONMENT\_lexSense, GRANULARITY\_lexSense, GRANULARIZATION\_lexSense, HUMAN--MACHINE--INTERFACE\_lexSense, INCIDENT--IMPACT--MANAGEMENT\_lexSense, INTERCHANGEABILITY\_lexSense, INTERFACE\_lexSense, INTEROPERABILITY\_lexSense, MAINTAINABILITY\_lexSense, MODULARITY\_lexSense, NARROW--INTERFACES\_lexSense, OPEX\_lexSense, PORTABILITY\_lexSense, REAL-TIME--CONFLICT--DETECTION\_lexSense, REUSABILITY\_lexSense, SCALABILITY\_lexSense, SECTIONAL--RUNTIME--CALCULATION\_lexSense, SSI\_lexSense, SUB-SYSTEM--SOMETIMES--CALLED--\_BUILDING--BLOCK\_lexSense, SWITCH\_lexSense, TERM--STATUS--DESCRIPTION--OPERATING--STATE--DRAFT--THE--OPERATING--STATE--DESCRIBES--THE\_lexSense, TESTABILITY\_lexSense, TMS--DAILY--TOPOLOGY\_lexSense, TOPOLOGY--MASTER--DATA--VALIDATION--AND--IMPORT\_lexSense, UPDATEABILITY\_lexSense, UPGRADEABILITY\_lexSense

## ontolex:LexicalConcept entities

### 687 ontolex:LexicalConcept in lex\_sp-defs-231113:

### 51 ontolex:LexicalConcept NEW from lex\_sp-defs-231012:

AB--ALLOCATION--BODY\_lexConcept, APPLICATION--GUIDELINE\_lexConcept, CI--COMMON--INTERFACE\_lexConcept, CMS--CAPACITY--MANAGEMENT--SYSTEM\_lexConcept, COMMON--STANDARD--PROPERTIES--OF--WORKITEMS\_lexConcept, CONFIGURATION--DATA\_lexConcept, CONTROL--COMMAND--AND--SIGNALLING\_lexConcept, DCM--DIGITAL--CAPACITY--MANAGEMENT\_lexConcept, DEF\_lexConcept, ECMT--EUROPEAN--CAPACITY--MANAGEMENT--TOOL\_lexConcept, EEE\_lexConcept, EMPTY--DEFINITION\_lexConcept, ENGINEERING--INPUT--DATA\_lexConcept, ERG-----EURO--RADIO--GATEWAY\_lexConcept, ERTMS--COMPRISES--OF--THE--EUROPEAN--TRAIN--CONTROL--SYSTEM--ETCS--I-E---A--CAB-SIGNALLING--SYSTEM--THAT--INCORPORATES--AUTOMATIC--TRAIN--PROTECTION--THE--GLOBAL--SYSTEM--FOR--MOBILE--COMMUNICATIONS--FOR--RAILWAYS--GSM-R--AND--OPERATING--RULES\_lexConcept, ESSENTIAL--FUNCTION\_lexConcept, ETA--ESTIMATED--TIME--OF--ARRIVAL\_lexConcept, EUROPE\_S--RAIL\_lexConcept, FAULT--CORRECTION--TIME\_lexConcept, FAULT--LOCALIZATION--TIME\_lexConcept, FUNCTION--CHECKOUT--TIME\_lexConcept, GUI--GRAPHICAL--USER--INTERFACE\_lexConcept, HMI--COMPONENT\_lexConcept, INFRASTRUCTURE--MANAGER\_lexConcept, MISSION--ASSIGNMENT\_lexConcept, NETWORK--COMPONENT\_lexConcept, NEW\_lexConcept, OPERATIONAL--MISSION--SEGMENT\_lexConcept, OPERATIONAL--MISSION\_lexConcept, OPERATIONAL--STOP\_lexConcept, PCS--PATH--COORDINATION--SYSTEM\_lexConcept, RAILWAY\_lexConcept, RELIABILITY--AVAILABILITY--MAINTAINABILITY--AND--SAFETY\_lexConcept, RELIABLE--DATA\_lexConcept, RIM--RAILWAY--INFRASTRUCTURE--MANAGER\_lexConcept, RIS--RAILWAY--INFRASTRUCTURE--SYSTEM\_lexConcept, ROAD--TRAFFIC--DRAFT\_lexConcept, ROC--RAILWAY--OPERATING--COMPANY\_lexConcept, ROLL--AWAY\_lexConcept, SECURE--COMMUNICATION\_lexConcept, SECURE--COMPONENT\_lexConcept, SHARED--SECURITY--SERVICES\_lexConcept, SR--AUTHORISATION\_lexConcept, SWITCH\_lexConcept\_5, TAF\_TAP--TSI--TECHNICAL--SPECIFICATION--FOR--INTEROPERABILITY--RELATING--TO--TELEMATICS\_lexConcept, TCR--TEMPORARY--CAPACITY--RESTRICTION\_lexConcept, TIMS-----TRAIN--INTEGRITY--MANAGEMENT--SYSTEM\_lexConcept, TMS--TRAFFIC--MANAGEMENT--SYSTEM\_lexConcept, TRAIN--RUNNING--NUMBER\_lexConcept, TRANSVERSAL--CCS--DOMAIN\_lexConcept, WIRELESS--COMPONENT\_lexConcept

### 48 ontolex:LexicalConcept REMOVED from lex\_sp-defs-231012:

AB--CI--CMS--DCM--ECMT--GUI--ETA--PCS--RIM--RIS--ROC--TAF\_TAP--TSI--TCR--TMS--ALLOCATION--BODY--C\_lexConcept, ADAPTABILITY\_lexConcept\_2, AUTOMATIC--CONFLICT--SOLUTION\_lexConcept\_2, CAPABILITY--MEANING\_lexConcept\_2, CAPACITY--PLAN--AND--DECISION--PROCESSING\_lexConcept\_2, CAPACITY--WASTE--MEANS\_lexConcept\_2, CAPACITY-FRIENDLY--BEHAVIOUR--MEANS\_lexConcept\_2, CAPEX\_lexConcept\_2, CHANGEABILITY\_lexConcept\_2, COMMON--STARNDARD--PROPERTIES--OF--WORKITEMS\_lexConcept, DEVIATION--DETECTION\_lexConcept\_2, ERTMS\_lexConcept\_3, EVOLVABILITY\_lexConcept\_2, EXCHANGEABILITY\_lexConcept\_2, FCKT\_lexConcept, FCT\_lexConcept, FFFIS-----FORM--FIT--FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept\_2, FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept\_3, FLT\_lexConcept, FUNCTIONAL--APPORTIONMENT\_lexConcept\_2, GRANULARITY\_lexConcept\_2, GRANULARIZATION\_lexConcept\_2, HUMAN--MACHINE--INTERFACE\_lexConcept\_3, IM-----INFRASTRUCTURE--MANAGER\_lexConcept, INCIDENT--IMPACT--MANAGEMENT\_lexConcept\_2, INTERCHANGEABILITY\_lexConcept\_2, INTERFACE\_lexConcept\_4, INTEROPERABILITY\_lexConcept\_3, MAINTAINABILITY\_lexConcept\_2, MITIGATION\_lexConcept, MODULARITY\_lexConcept\_4, NARROW--INTERFACES\_lexConcept\_2, OPEX\_lexConcept\_2, PORTABILITY\_lexConcept\_2, REAL-TIME--CONFLICT--DETECTION\_lexConcept\_2, REUSABILITY\_lexConcept\_2, SCALABILITY\_lexConcept\_2, SECTIONAL--RUNTIME--CALCULATION\_lexConcept\_2, SSI\_lexConcept\_2, SUB-SYSTEM--SOMETIMES--CALLED--\_BUILDING--BLOCK\_lexConcept\_2, TARGETSYSTEM---A--SUBSYSTEM--OF--THE--SYSTEM--PILLAR\_S--ARCHITECTURE--WHICH--IS--A--TARGET\_lexConcept, TERM--STATUS--DESCRIPTION--OPERATING--STATE--DRAFT--THE--OPERATING--STATE--DESCRIBES--THE\_lexConcept\_2, TESTABILITY\_lexConcept\_2, TIMS-----TRAIN--INTEGRITY\_lexConcept, TMS--DAILY--TOPOLOGY\_lexConcept\_2, TOPOLOGY--MASTER--DATA--VALIDATION--AND--IMPORT\_lexConcept\_2, UPDATEABILITY\_lexConcept\_2, UPGRADEABILITY\_lexConcept\_2

### 189 ontolex:LexicalConcept MODIFIED from lex\_sp-defs-231012:

ACCIDENT\_lexConcept, ACTOR--VARIANT\_lexConcept, APPLICATION\_lexConcept, ARCHITECTURAL--CONCEPT\_lexConcept, ASFA\_lexConcept, ATO\_lexConcept, AUTHENTICATION\_lexConcept, AVAILABILITY--%3COF--A--PRODUCT%3E\_lexConcept, BACKWARDS--COMPATIBILITY\_lexConcept, BASIC--INTEGRITY--PLATFORM--INDEPENDENCE--INTERFACE--I4\_lexConcept, BUILDING--BLOCK--CONFIGURATION--MANIFEST\_lexConcept, BUILDING--BLOCK\_lexConcept, BUILDING--STRATEGY\_lexConcept, CAPACITY--WASTE--MEANS\_lexConcept, CAPACITY-FRIENDLY--BEHAVIOUR--MEANS\_lexConcept, CAPELLA--VIEWPOINT\_lexConcept, CAPEX\_lexConcept, CBM\_lexConcept, CCF\_lexConcept, CCS--CONFIGURATION--MANIFEST\_lexConcept, CCS--CONFIGURATION\_lexConcept, CCS--FEATURE\_lexConcept, CCS\_TMS--DATA--MODEL\_lexConcept, CMS\_lexConcept, CM\_lexConcept, COMPONENT\_lexConcept, CONCEPTUAL--GLOSSARY\_lexConcept, CONDITION--MONITORING--%3COF--AN--ITEM%3E\_lexConcept, CONFIDENTIALITY\_lexConcept, CONFIGURATION--ITEM\_lexConcept, CONFIGURATION\_lexConcept, CONSOLIDATED--GLOSSARY\_lexConcept, CORRECTIVE--MAINTENANCE\_lexConcept, DATA--MODEL--LAYER\_lexConcept, DEFINITION\_lexConcept\_2, DEVELOPMENT--TASK\_lexConcept, DEVICE\_lexConcept, DISTRIBUTION--JOB\_lexConcept, ERTMS\_lexConcept, ERTMS\_lexConcept\_2, ETCS\_lexConcept\_2, EU-RAILGOVERNING--BOARD\_lexConcept, EVOLVABILITY\_lexConcept, EXCHANGEABILITY\_lexConcept, FAIL-SAFE\_lexConcept, FAIL-SAFE\_lexConcept\_2, FAILURE--CAUSE\_lexConcept, FAILURE--MODE\_lexConcept, FAILURE--RATE--\_821-12-21\_lexConcept, FAILURE--RATE\_lexConcept, FAILURE--RATE\_lexConcept\_2, FAILURE\_lexConcept, FAULT--%3COF--AN--ITEM%3E\_lexConcept, FAULT--DETECTION--TIME\_lexConcept, FAULT--TREE\_lexConcept, FEATURE--SUMMARY\_lexConcept, FFFIS-----FORM--FIT--FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept, FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept\_2, FMECA\_lexConcept, FMECA\_lexConcept\_2, FOLLOW--A--TRACE\_lexConcept, FTA\_lexConcept, FULL--BACKWARDS--COMPATIBILITY\_lexConcept, FUNCTIONAL--APPLICATION\_lexConcept, FUNCTIONAL--APPORTIONMENT\_lexConcept, FUNCTIONAL--TEAM\_lexConcept, FUNCTION\_lexConcept, FUNKTIONAL--TEAM\_lexConcept, GENERIC--WORKFLOW--TYPES\_lexConcept, GRANULARITY\_lexConcept, GROUND--FOOTPRINT\_lexConcept, HAZARD--MITIGATION\_lexConcept, HAZARD\_lexConcept, HAZARD\_lexConcept\_2, HAZOP\_lexConcept, HFI--ACTIVITIES\_lexConcept, HSI\_lexConcept, HUMAN--FACTORS\_lexConcept, HUMAN--MACHINE--INTERFACE\_lexConcept\_2, IMPERSONATION\_lexConcept, INPUT--DOCUMENTS\_lexConcept, INTEGRATION--TASK\_lexConcept, INTERCHANGEABILITY\_lexConcept, INTERFACE\_lexConcept\_2, INTERFACE\_lexConcept\_3, INTEROPERABILITY\_lexConcept, INTEROPERABILITY\_lexConcept\_2, LEXICAL--GLOSSARY\_lexConcept, LOGIAL--COMPONENT\_lexConcept, LOGISTIC--DELAY\_lexConcept, MACMT\_lexConcept, MAD\_lexConcept, MAINTAINABILITY--%3COF--AN--ITEM%3E\_lexConcept, MAINTAINABILITY\_lexConcept, MANAGEMENT--BY--TRACEABILITY--KANBAN--BASED\_lexConcept, MDBF\_lexConcept, MDBSF\_lexConcept, METHODOLOGY\_lexConcept, MFDT\_lexConcept, MLD\_lexConcept, MODE--TRANSITION\_lexConcept, MODEL--SYNCHRONISATION\_lexConcept, MODULARITY\_lexConcept, MODULARITY\_lexConcept\_2, MODULE\_lexConcept, MOE\_lexConcept, MOTBF\_lexConcept, MRT\_lexConcept, MTBF\_lexConcept\_2, MTBSF\_lexConcept, MTD\_lexConcept, MTTFF\_lexConcept, MTTF\_lexConcept, MTTR\_lexConcept, NATIONAL--IMPLEMENATION--PLAN\_lexConcept, NON-FUNCTIONAL--REQUIREMENTS\_lexConcept, NOTIF-IT\_lexConcept, OAB\_--\_ENTITIES\_ACTORS\_lexConcept, OAB\_--\_PROCESS--INVOLVEMENT\_lexConcept, OAIB\_--\_OPERATIONAL--PROCESS\_lexConcept, OBJECTIVES\_lexConcept, OCB\_--\_INVOLVEMENT\_lexConcept, OCB\_--\_STRUCTURE\_lexConcept, OES\_--\_OPERATIONAL--PROCESS\_--\_SCENARIO\_lexConcept, OPD\_--\_OPERATIONAL--PROCESS\_--\_LOGIC\_lexConcept, OPERATING--TIME--TO--FAILURE--%3COF--AN--ITEM%3E\_lexConcept, OPERATING--TIME\_lexConcept, OPERATIONAL--HARMONIZATION\_lexConcept, ORS--OPERATIONAL--REQUIREMENT--SPECIFICATION\_lexConcept, PERFORMANCE--%3C--OF--AN--ITEM--%3E\_lexConcept, PFH\_lexConcept, PFH\_lexConcept\_2, PFH\_lexConcept\_3, PHYSICAL--LINK\_lexConcept, PHYSICAL--PORT\_lexConcept, PLATEAU\_lexConcept, RBD\_lexConcept, RELIABILITY--%3COF--AN--ITEM%3E\_lexConcept, REQUIREMENT--STATEMENT\_lexConcept, RISK--%3COF--A--HAZARD%3E\_lexConcept, RISK--ANALYSIS\_lexConcept, RISK--ASSESSMENT\_lexConcept, RISK--EVALUATION\_lexConcept, SAFE--STATE--\_821-12-49\_lexConcept, SAFE--STATE\_lexConcept, SAFE--STATE\_lexConcept\_2, SAFETY--INVARIANT\_lexConcept, SEMP--REQUIREMENTS--TYPES\_lexConcept, SERA-CCS\_lexConcept, SERIOUS--ACCIDENT\_lexConcept, SITUATION\_lexConcept, SPECIFICATION--TASK\_lexConcept, SRD\_lexConcept, STAKEHOLDER--NEEDS\_lexConcept, STAKEHOLDER\_lexConcept, STPA\_lexConcept, SUB-SYSTEM--SOMETIMES--CALLED--\_BUILDING--BLOCK\_lexConcept, SUBSYSTEM\_lexConcept, SUC\_lexConcept, SWITCH\_lexConcept\_2, SWITCH\_lexConcept\_4, SYSTEM--AND--INNOVATION--PROGRAMME--BOARD\_lexConcept, SYSTEM--DEVELOPMENT--LIFE--CYCLE\_lexConcept, SYSTEM--LEVELS\_lexConcept, SYSTEM--OF--SYSTEMS\_lexConcept, SYSTEM--PILLAR--CORE--GROUP\_lexConcept, SYSTEM--PILLAR--DELIVERABLES--\_OUTPUT--DOCUMENTS\_lexConcept, SYSTEM--PILLAR--STEERING--GROUP\_lexConcept, SYSTEM--PILLAR--UNIT--CHAIRS--THE--SYSTEM--PILLAR--CORE--GROUP\_lexConcept, SYSTEM--REQUIREMENTS\_lexConcept, SYSTEM\_lexConcept, TAILORING--OF--REQUIREMENT--BREAKDOWN\_lexConcept, TARGET--PICTURE\_lexConcept, TECHNICAL--DELAY\_lexConcept, TEMPORARY--SHUNTING--AREA--TSHA--OR-DEF-160--TEMPORARY--SHUNTING--AREA--OR-DEF-161--DEF\_lexConcept, TERM--STATUS--DESCRIPTION--OPERATING--STATE--DRAFT--THE--OPERATING--STATE--DESCRIBES--THE\_lexConcept, TESTABILITY\_lexConcept, THE--\_TRACE\_--FOR--A--WORK--ITEM--CHAIN\_TREE\_GRAPH\_lexConcept, TLS\_lexConcept, TRACK--FOOTPRINT\_lexConcept, TRADE-SPACE--FACTOR\_lexConcept, TRAIN-CENTRIC--TRACK--OCCUPANCY\_lexConcept, UPLINKING\_lexConcept, WORK--ITEM--EDITOR\_lexConcept, WORK--ITEM\_lexConcept\_2, WORKFLOW--AND--WORKFLOW--RULES\_lexConcept, WORKFLOW--PRIORITISATION--STRATEGY--TO--BE--DECIDED--PER--AREA\_lexConcept, WORKFLOW--STEP--ON--STEP--IN--A--WORKITEM--TRACE\_lexConcept, WORKSTEP--\_WORKITEM--CHECK\_lexConcept

# Modified Entities

## lexinfo:AbbreviatedForm entities

 ➱ No modification occured in this type of Entities

## ontolex:LexicalEntry entities

 ➱ No modification occured in this type of Entities

## ontolex:Form entities

 ➱ No modification occured in this type of Entities

## ontolex:LexicalSense entities

### ontorail:ontolex:LexicalSense 0 cosmetic changes have been skipped

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:ADAPTABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :ADAPTABILITY\_lexConcept, -- :ADAPTABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:AUTOMATIC--CONFLICT--SOLUTION\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :AUTOMATIC--CONFLICT--SOLUTION\_lexConcept, -- :AUTOMATIC--CONFLICT--SOLUTION\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:CAPABILITY--MEANING\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :CAPABILITY--MEANING\_lexConcept, -- :CAPABILITY--MEANING\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:CAPACITY--PLAN--AND--DECISION--PROCESSING\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :CAPACITY--PLAN--AND--DECISION--PROCESSING\_lexConcept, -- :CAPACITY--PLAN--AND--DECISION--PROCESSING\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:CAPACITY--WASTE--MEANS\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :CAPACITY--WASTE--MEANS\_lexConcept, -- :CAPACITY--WASTE--MEANS\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:CAPACITY-FRIENDLY--BEHAVIOUR--MEANS\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :CAPACITY-FRIENDLY--BEHAVIOUR--MEANS\_lexConcept, -- :CAPACITY-FRIENDLY--BEHAVIOUR--MEANS\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:CAPEX\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :CAPEX\_lexConcept, -- :CAPEX\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:CHANGEABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :CHANGEABILITY\_lexConcept, -- :CHANGEABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:DEVIATION--DETECTION\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :DEVIATION--DETECTION\_lexConcept, -- :DEVIATION--DETECTION\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:ERTMS\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :ERTMS\_lexConcept, :ERTMS\_lexConcept\_2, -- :ERTMS\_lexConcept\_3

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:EVOLVABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :EVOLVABILITY\_lexConcept, -- :EVOLVABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:EXCHANGEABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :EXCHANGEABILITY\_lexConcept, -- :EXCHANGEABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:FFFIS-----FORM--FIT--FUNCTIONAL--INTERFACE--SPECIFICATION\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :FFFIS-----FORM--FIT--FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept, -- :FFFIS-----FORM--FIT--FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept, :FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept\_2, -- :FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept\_3

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:FUNCTIONAL--APPORTIONMENT\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :FUNCTIONAL--APPORTIONMENT\_lexConcept, -- :FUNCTIONAL--APPORTIONMENT\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:GRANULARITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :GRANULARITY\_lexConcept, -- :GRANULARITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:GRANULARIZATION\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :GRANULARIZATION\_lexConcept, -- :GRANULARIZATION\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:HUMAN--MACHINE--INTERFACE\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :HUMAN--MACHINE--INTERFACE\_lexConcept, :HUMAN--MACHINE--INTERFACE\_lexConcept\_2, -- :HUMAN--MACHINE--INTERFACE\_lexConcept\_3

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:INCIDENT--IMPACT--MANAGEMENT\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :INCIDENT--IMPACT--MANAGEMENT\_lexConcept, -- :INCIDENT--IMPACT--MANAGEMENT\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:INTERCHANGEABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :INTERCHANGEABILITY\_lexConcept, -- :INTERCHANGEABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:INTERFACE\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :INTERFACE\_lexConcept, :INTERFACE\_lexConcept\_2, :INTERFACE\_lexConcept\_3, -- :INTERFACE\_lexConcept\_4

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:INTEROPERABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :INTEROPERABILITY\_lexConcept, :INTEROPERABILITY\_lexConcept\_2, -- :INTEROPERABILITY\_lexConcept\_3

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:MAINTAINABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :MAINTAINABILITY\_lexConcept, -- :MAINTAINABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:MODULARITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :MODULARITY\_lexConcept, :MODULARITY\_lexConcept\_2, :MODULARITY\_lexConcept\_3, -- :MODULARITY\_lexConcept\_4

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:NARROW--INTERFACES\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :NARROW--INTERFACES\_lexConcept, -- :NARROW--INTERFACES\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:OPEX\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :OPEX\_lexConcept, -- :OPEX\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:PORTABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :PORTABILITY\_lexConcept, -- :PORTABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:REAL-TIME--CONFLICT--DETECTION\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :REAL-TIME--CONFLICT--DETECTION\_lexConcept, -- :REAL-TIME--CONFLICT--DETECTION\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:REUSABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :REUSABILITY\_lexConcept, -- :REUSABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:SCALABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :SCALABILITY\_lexConcept, -- :SCALABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:SECTIONAL--RUNTIME--CALCULATION\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :SECTIONAL--RUNTIME--CALCULATION\_lexConcept, -- :SECTIONAL--RUNTIME--CALCULATION\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:SSI\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :SSI\_lexConcept, -- :SSI\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:SUB-SYSTEM--SOMETIMES--CALLED--\_BUILDING--BLOCK\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :SUB-SYSTEM--SOMETIMES--CALLED--\_BUILDING--BLOCK\_lexConcept, -- :SUB-SYSTEM--SOMETIMES--CALLED--\_BUILDING--BLOCK\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:SWITCH\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :SWITCH\_lexConcept, :SWITCH\_lexConcept\_2, :SWITCH\_lexConcept\_3, :SWITCH\_lexConcept\_4, ++ :SWITCH\_lexConcept\_5

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:TERM--STATUS--DESCRIPTION--OPERATING--STATE--DRAFT--THE--OPERATING--STATE--DESCRIBES--THE\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :TERM--STATUS--DESCRIPTION--OPERATING--STATE--DRAFT--THE--OPERATING--STATE--DESCRIBES--THE\_lexConcept, -- :TERM--STATUS--DESCRIPTION--OPERATING--STATE--DRAFT--THE--OPERATING--STATE--DESCRIBES--THE\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:TESTABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :TESTABILITY\_lexConcept, -- :TESTABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:TMS--DAILY--TOPOLOGY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :TMS--DAILY--TOPOLOGY\_lexConcept, -- :TMS--DAILY--TOPOLOGY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:TOPOLOGY--MASTER--DATA--VALIDATION--AND--IMPORT\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :TOPOLOGY--MASTER--DATA--VALIDATION--AND--IMPORT\_lexConcept, -- :TOPOLOGY--MASTER--DATA--VALIDATION--AND--IMPORT\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:UPDATEABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :UPDATEABILITY\_lexConcept, -- :UPDATEABILITY\_lexConcept\_2

### ontorail:ontolex:LexicalSense lex\_sp-defs-231113:UPGRADEABILITY\_lexSense modifications from lex\_sp-defs-231012:

 == ontolex:isLexicalizedSenseOf => :UPGRADEABILITY\_lexConcept, -- :UPGRADEABILITY\_lexConcept\_2

## ontolex:LexicalConcept entities

### ontorail:ontolex:LexicalConcept 0 cosmetic changes have been skipped

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ACCIDENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "‘accident’ means an unwanted or unintended sudden event or a specific chain of such events which have harmful consequences; accidents are divided into the following categories: collisions; derailments; level crossing accidents; accidents to persons involving rolling stock in motion; fires and others;
\n[SOURCE: SPPRAMSS-337 - [Directive (EU) 2016/798] Article 3 Definitions (11) ]", -- "‘accident’ means an unwanted or unintended sudden event or a specific chain of such events which have harmful consequences; accidents are divided into the following categories: collisions; derailments; level crossing accidents; accidents to persons involving rolling stock in motion; fires and others;\n\n[SOURCE: SPPRAMSS-337 - [Directive (EU) 2016/798] Article 3 Definitions (11) ]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ACTOR--VARIANT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "An actor variant is an deviation to an existing actor regarding function deployment.
\nIt arises in two cases:
\n\* An additional (unusual) function is deployed to the actor variant.
\n\* A common actor function is removed and deployed to another actor or to a system.
\nIdentification of actor variants is a precondition for harmonisation of operational processes.", -- "An actor variant is an deviation to an existing actor regarding function deployment.\n\nIt arises in two cases:\n\n\* An additional (unusual) function is deployed to the actor variant.\n\n\* A common actor function is removed and deployed to another actor or to a system.\n\nIdentification of actor variants is a precondition for harmonisation of operational processes."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:APPLICATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "An application in SP context is the resulting situation of a system beeing used in a specific environment by specific actors.
\nIn short - Application = System + Processes + Actors + Environment.", -- "An application in SP context is the resulting situation of a system beeing used in a specific environment by specific actors. \n\nIn short - Application = System + Processes + Actors + Environment."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ARCHITECTURAL--CONCEPT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The architectural process comprises four steps, each dealing with a separate concern.
\nThe general concept implements the architecture recommendations from the System Pillar report [SPREP, page 11] for a function-based architecture and a layered architecture approach. Both concepts can be realised with the architectural principles described herein.These steps are described in detail in the following chapters.
\n\* Operational analysis (OA): identify the operational process needs that are to be supported by the technical systems.{comment:4} This analysis should focus as purely as possible on the processes and ideally does not take any specific technical system architecture into account. The operational analysis is usually performed on an abstraction layer above the topmost system in the systems of systems hierarchy and performed only once. See Chapter 5.6
\n\* System analysis (SA): identify the needs of the system of interest. This step does not design a specific technical solution but captures the needs for the future system. It hence represents a statement of work and not a finished piece of engineering. It is used to rationalize the decision, which operational processes will be performed by the system of interest, and which will be not (these processes then mostly will be either performed by other systems or by human actors and defined as operating rules). System analysis is performed recursively:
\n \* Once for the topmost system of systems, deriving the initial need from the operational analysis
\n \* Multiple times for each system of system decomposition step, deriving the system needs of the lower level of decomposition from the higher level of decomposition
\n\* Logical architecture (LA): design a solution to the system needs based on solution concepts and architectural concepts. Split the system functions based on solution concepts (e.g. absolute positioning vs reference point based localisation, moving blocks, fixed blocks or hybrid) so that it becomes clear, how and by which steps the inputs to a system function are converted to the outputs. This step does not yet define an architecture and does not refer to technical solution concepts like ETCS or ATO. As the system under consideration is still a blackbox, the logical architecture still leaves the question open, what subsystem structure is the to be used (e.g. very modular subsystems vs. bigger subsystems or combined HW/SW subsystems vs. SW-modules on a common platform). This step is performed once, before the subsystem architecture shall be derived.
\n\* Subsystem architecture (SSA): design the final set of tenderable subsystems and integrate all necessary non-functional requirements. This step integrates all considerations on the intended structure of subsystems and interfaces (down to FFFIS) as well as all open technical aspects into a consistent architectural definition.{comment:3}", -- "The architectural process comprises four steps, each dealing with a separate concern.\n\nThe general concept implements the architecture recommendations from the System Pillar report [SPREP, page 11] for a function-based architecture and a layered architecture approach. Both concepts can be realised with the architectural principles described herein.These steps are described in detail in the following chapters. \n\n\* Operational analysis (OA): identify the operational process needs that are to be supported by the technical systems.{comment:4} This analysis should focus as purely as possible on the processes and ideally does not take any specific technical system architecture into account. The operational analysis is usually performed on an abstraction layer above the topmost system in the systems of systems hierarchy and performed only once. See Chapter 5.6\n\n\* System analysis (SA): identify the needs of the system of interest. This step does not design a specific technical solution but captures the needs for the future system. It hence represents a statement of work and not a finished piece of engineering. It is used to rationalize the decision, which operational processes will be performed by the system of interest, and which will be not (these processes then mostly will be either performed by other systems or by human actors and defined as operating rules). System analysis is performed recursively:\n\n \* Once for the topmost system of systems, deriving the initial need from the operational analysis\n\n \* Multiple times for each system of system decomposition step, deriving the system needs of the lower level of decomposition from the higher level of decomposition\n\n\* Logical architecture (LA): design a solution to the system needs based on solution concepts and architectural concepts. Split the system functions based on solution concepts (e.g. absolute positioning vs reference point based localisation, moving blocks, fixed blocks or hybrid) so that it becomes clear, how and by which steps the inputs to a system function are converted to the outputs. This step does not yet define an architecture and does not refer to technical solution concepts like ETCS or ATO. As the system under consideration is still a blackbox, the logical architecture still leaves the question open, what subsystem structure is the to be used (e.g. very modular subsystems vs. bigger subsystems or combined HW/SW subsystems vs. SW-modules on a common platform). This step is performed once, before the subsystem architecture shall be derived.\n\n\* Subsystem architecture (SSA): design the final set of tenderable subsystems and integrate all necessary non-functional requirements. This step integrates all considerations on the intended structure of subsystems and interfaces (down to FFFIS) as well as all open technical aspects into a consistent architectural definition.{comment:3}"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ASFA\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ Anuncio de Señales y Frenado Automático (ASFA; "Announcement of Signals and Automatic Braking");
\n Is an Automatic Protetion System which is widely used on the Spanish rail network. It consists of a mechanism that stops a train if the driver does not properly heed signals. """, -- """ Anuncio de Señales y Frenado Automático (ASFA; "Announcement of Signals and Automatic Braking"); \n\n Is an Automatic Protetion System which is widely used on the Spanish rail network. It consists of a mechanism that stops a train if the driver does not properly heed signals. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ATO\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Automatic Train Operation is technology for automating the operation of trains. The degree of the automatisation is shown by the Grade of Automatation (GoA).
\nGoA0: train operating on-sight, no automation
\nGoA1: train operating manual, train driver controls starting, stopping, passenger service functions as opening and closing doors and handling emergency. Train protection systems like ETCS L1 in place.
\nGoA2: train operating semi-automatic. Starting and stopping automated using advanced train protection systems like ETCS L2 or 3, driver operates passenger service functions and handles emergencies
\nGoA3: driverless train operation. Starting and stopping automated, service staff operates passenger service functions and handles emergencies
\nGoA4: unattended train operation. All operations are fully automated without any on-train staff", -- "Automatic Train Operation is technology for automating the operation of trains. The degree of the automatisation is shown by the Grade of Automatation (GoA).\n\nGoA0: train operating on-sight, no automation\n\nGoA1: train operating manual, train driver controls starting, stopping, passenger service functions as opening and closing doors and handling emergency. Train protection systems like ETCS L1 in place.\n\nGoA2: train operating semi-automatic. Starting and stopping automated using advanced train protection systems like ETCS L2 or 3, driver operates passenger service functions and handles emergencies\n\nGoA3: driverless train operation. Starting and stopping automated, service staff operates passenger service functions and handles emergencies\n\nGoA4: unattended train operation. All operations are fully automated without any on-train staff"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:AUTHENTICATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The process to verify the identity of communicating peers.
\n (source: SPPRAMSS-1705 - [UNSIG Subset-146] )", -- "The process to verify the identity of communicating peers.\n\n (source: SPPRAMSS-1705 - [UNSIG Subset-146] )"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:AVAILABILITY--%3COF--A--PRODUCT%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Ability of an item to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval, assuming that the required external resources are provided
\n[SOURCE: IEC 60050-821: FDIS2016, 821-05-82, modified]
\nSource: SPPRAMSS-349 - [EN 50126-1:2017]", -- "Ability of an item to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval, assuming that the required external resources are provided \n\n[SOURCE: IEC 60050-821: FDIS2016, 821-05-82, modified]\n\nSource: SPPRAMSS-349 - [EN 50126-1:2017]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:BACKWARDS--COMPATIBILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ Backwards compatibility is a design or software feature that allows a product, system, or software application to remain compatible with earlier versions or older hardware.
\n
\n In the context of software, it means that a newer version of a program can still run and interact with files, data, or systems that were created using previous versions of that software. This ensures that users can upgrade to the latest version without losing access to their existing data or needing to make significant changes to their workflows. It's a valuable feature that enhances user convenience and reduces disruptions when technology evolves.
\n
\n Backward compatibility for hardware, also known as "hardware backward compatibility," refers to the ability of a newer piece of hardware to work seamlessly with older hardware components or peripherals, such as connectors, interfaces, or accessories. In the context of hardware, backward compatibility typically ensures that the new hardware can accommodate and interact with devices or components that were designed for older hardware specifications.
\n Hardware backward compatibility is important for user convenience, cost savings, and reducing the need for immediate updates to all associated hardware components when a single component is upgraded. It often requires the inclusion of older ports or connectors on newer hardware or the development of adapters or converters to bridge the compatibility gap between old and new technologies. """, -- """ Backwards compatibility is a design or software feature that allows a product, system, or software application to remain compatible with earlier versions or older hardware.\n\n\n\n In the context of software, it means that a newer version of a program can still run and interact with files, data, or systems that were created using previous versions of that software. This ensures that users can upgrade to the latest version without losing access to their existing data or needing to make significant changes to their workflows. It's a valuable feature that enhances user convenience and reduces disruptions when technology evolves.\n\n\n\n Backward compatibility for hardware, also known as "hardware backward compatibility," refers to the ability of a newer piece of hardware to work seamlessly with older hardware components or peripherals, such as connectors, interfaces, or accessories. In the context of hardware, backward compatibility typically ensures that the new hardware can accommodate and interact with devices or components that were designed for older hardware specifications.\n\n Hardware backward compatibility is important for user convenience, cost savings, and reducing the need for immediate updates to all associated hardware components when a single component is upgraded. It often requires the inclusion of older ports or connectors on newer hardware or the development of adapters or converters to bridge the compatibility gap between old and new technologies. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:BASIC--INTEGRITY--PLATFORM--INDEPENDENCE--INTERFACE--I4\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Interface I4 provides a platform independence API for functional applications in the non-safety area.", -- "Interface I4 provides a basic integrity (SIL0) platform independence API for functional applications."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:BUILDING--BLOCK--CONFIGURATION--MANIFEST\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The Building Block (BB) Configuration Manifest is a file used to describe the configuration items of a BB Configuration. The BB Configuration Manifest file lists all configuration items of a specific building block, it is a generic definition (before integration in its environment) of the building block of a specific supplier.
\n Note: the actual BB Configuration Items are not part of the BB Configuration Manifest, they are only referenced in the BB Configuration Manifest file.", -- "The Building Block (BB) Configuration Manifest is a file used to describe the configuration items of a BB Configuration. The BB Configuration Manifest file lists all configuration items of a specific building block, it is a generic definition (before integration in its environment) of the building block of a specific supplier.\n\n Note: the actual BB Configuration Items are not part of the BB Configuration Manifest, they are only referenced in the BB Configuration Manifest file."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:BUILDING--BLOCK\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "{comment:33}
\nA building block is a sourceable unit of the CCS system (hardware and/or software), having standardised functionality, standardised performance (RAM), standardised safety (including Tolerable Functional Failure Rate [TFFR], Safety Integrity Level [SIL] and Safety Related Application Conditions [SRAC]), standardised security and standardised interfaces towards other building blocks and/or external systems.
\nA building block can also be defined as a standardised module. {comment:158}", -- "{comment:33}\n\nA building block is a sourceable unit of the CCS system (hardware and/or software), having standardised functionality, standardised performance (RAM), standardised safety (including Tolerable Functional Failure Rate [TFFR], Safety Integrity Level [SIL] and Safety Related Application Conditions [SRAC]), standardised security and standardised interfaces towards other building blocks and/or external systems.\n\nA building block can also be defined as a standardised module. {comment:158}"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:BUILDING--STRATEGY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "processus allowing to regroup physical components into consistent sub-collection in terms of industral development
\n concerns and a strategy point of view", -- "processus allowing to regroup physical components into consistent sub-collection in terms of industral development \n\n concerns and a strategy point of view"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CAPACITY--WASTE--MEANS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Paths are ordered without proper need, allocated and modified after allocation so that other ROCs cannot sensibly make use of them
\n\* Allocated paths are changed by RIMs after allocation due to the impact of TCRs by RIMs (same one allocating the path or another one on a different network)
\n\* An unnecessary number of paths are not allocated or are cancelled by RIMs due to poor alignment of several planned TCRs
\n\* Paths are not allocated or are cancelled by RIMs due to planned TCRs at times when little or no work is actually being carried out
\n\* Paths are not allocated or are cancelled by RIMs due to planned TCRs that were cancelled
\n\* Paths are not usable (or re-usable) due to the changes being communicated too late (i.e., due to paths being cancelled too late, TCRs cancelled, etc.)", -- "\* Paths are ordered without proper need, allocated and modified after allocation so that other ROCs cannot sensibly make use of them\n\n\* Allocated paths are changed by RIMs after allocation due to the impact of TCRs by RIMs (same one allocating the path or another one on a different network)\n\n\* An unnecessary number of paths are not allocated or are cancelled by RIMs due to poor alignment of several planned TCRs\n\n\* Paths are not allocated or are cancelled by RIMs due to planned TCRs at times when little or no work is actually being carried out\n\n\* Paths are not allocated or are cancelled by RIMs due to planned TCRs that were cancelled\n\n\* Paths are not usable (or re-usable) due to the changes being communicated too late (i.e., due to paths being cancelled too late, TCRs cancelled, etc.)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CAPACITY-FRIENDLY--BEHAVIOUR--MEANS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* A path confirmed is a commitment from both ROC and RIM, which should be kept after it has been allocated and accepted
\n\* Ordering capacity only when the market need for it need has been confirmed (ROC)
\n\* Give back the capacity as soon the market need changes, or other external requirements are known (ROC)
\n\* Defining the capacity for TCRs in a way that no more capacity than necessary is blocked – in dialogue between ROCs and RIMs
\n\* Planning the majority of TCRs and their impact to paths before path allocation (RIM)
\n\* Reducing TCRs in operational timetable to the minimum and planning their impact to paths/trains as early as possible (RIM)
\n\* Stabilising TCR planning - avoiding TCR cancellations that lead to a double loss of business, as well as massive re-planning of scheduled TCRs, which results in high efforts in operational planning (RIM)", -- "\* A path confirmed is a commitment from both ROC and RIM, which should be kept after it has been allocated and accepted\n\n\* Ordering capacity only when the market need for it need has been confirmed (ROC)\n\n\* Give back the capacity as soon the market need changes, or other external requirements are known (ROC)\n\n\* Defining the capacity for TCRs in a way that no more capacity than necessary is blocked – in dialogue between ROCs and RIMs\n\n\* Planning the majority of TCRs and their impact to paths before path allocation (RIM)\n\n\* Reducing TCRs in operational timetable to the minimum and planning their impact to paths/trains as early as possible (RIM)\n\n\* Stabilising TCR planning - avoiding TCR cancellations that lead to a double loss of business, as well as massive re-planning of scheduled TCRs, which results in high efforts in operational planning (RIM)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CAPELLA--VIEWPOINT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A viewpoint is the tailoring of a capella diagram:
\n
\n\* It has a specific purpose
\n\* It follows some diagram rules, that enforce this purpose
\n\* It is exported from Capella to Polarion in graphical form with a pre-defined filter", -- "A viewpoint is the tailoring of a capella diagram:\n\n \n\n\* It has a specific purpose\n\n\* It follows some diagram rules, that enforce this purpose\n\n\* It is exported from Capella to Polarion in graphical form with a pre-defined filter"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CAPEX\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capital Expenditures or Capital Expenses", -- "Capital Expenditures oder Capital Expenses"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CBM\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "condition-based maintenance
\npreventive maintenance based on the assessment of physical condition
\nNote 1 to entry: The condition assessment may be by operator observation, conducted according to a schedule, or by condition monitoring (192-06-28 SPPRAMSS-4462 - condition monitoring, <of an item> ) of system parameters.
\n[SOURCE: IEC 60050-192:2015, 192-06-07]", -- "condition-based maintenance\n\npreventive maintenance based on the assessment of physical condition\n\nNote 1 to entry: The condition assessment may be by operator observation, conducted according to a schedule, or by condition monitoring (192-06-28 SPPRAMSS-4462 - condition monitoring, <of an item> ) of system parameters.\n\n[SOURCE: IEC 60050-192:2015, 192-06-07]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CCF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ Common Cause Failures: failures of multiple items, which would otherwise be considered independent of one another, resulting from a single cause
\n
\n Note 1 to entry: Common cause failures can also be "common mode failures" (IEV 192-03-19).
\n Note 2 to entry: The potential for common cause failures reduces the effectiveness of system redundancy.
\n[SOURCE: IEC 60050-192:2015, 192-03-18] """, -- """ Common Cause Failures: failures of multiple items, which would otherwise be considered independent of one another, resulting from a single cause\n\n \n\n Note 1 to entry: Common cause failures can also be "common mode failures" (IEV 192-03-19).\n\n Note 2 to entry: The potential for common cause failures reduces the effectiveness of system redundancy.\n\n[SOURCE: IEC 60050-192:2015, 192-03-18] """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CCS--CONFIGURATION--MANIFEST\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The CCS Configuration Manifest is a file used to describe the configuration items of a CCS Configuration. The CCS Configuration Manifest file lists all field components of a specific CCS instance deployment, it is an overarching definition of the specific CCS instance deployment. By means of the CCS Configuration Manifest the integrity of a specific CCS instance deployment can be ensured.
\n Note: the actual CCS Configuration Items are not part of the CCS Configuration Manifest, they are only referenced in the CCS Configuration Manifest file.", -- "The CCS Configuration Manifest is a file used to describe the configuration items of a CCS Configuration. The CCS Configuration Manifest file lists all field components of a specific CCS instance deployment, it is an overarching definition of the specific CCS instance deployment. By means of the CCS Configuration Manifest the integrity of a specific CCS instance deployment can be ensured.\n\n Note: the actual CCS Configuration Items are not part of the CCS Configuration Manifest, they are only referenced in the CCS Configuration Manifest file."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CCS--CONFIGURATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The CCS Configuration is an exhaustive, unambiguous description (all configuration items and an overarching CCS manifest file) of all building blocks necessary to operate a physical instance of a CCS System, approved (in line with the CENELEC process) for deployment to a specific CCS Instance (identified via an identifier, for instance the already existing NID\_ENGINE, or other identifier already in use) and authorised to either operate on specific Railway Networks (identified via unique Railway Network Identifiers) or to be part of a specific Railway Network.
\nThe CCS Configuration is used for distribution of statical or semi-statical data (update data packet). It does not cover dynamic operational data like Moving Authority (for ETCS) or Journey Profiles (for ATO). The CCS Configuration is intended for CCS building blocks being present in multiple units, either close to infrastructure elements (e.g. CCS trackside instance and related signalling components like object controller) or CCS components on-board a train. It is not the aim to use CCS configuration management process (and the related CCS Configuration) for the update of large cloud based applications (installed on server farms). For the latter there are more suitable solutions from the IT industry.
\nIt is not a prerequisite that CCS building blocks need to stay in operation when activating / installing a CCS Configuration (however, this is allowed). These components could shut down / restart, basically go out of operation, or remain in operation.
\nThe main sources for the CCS Configuration (configuration items and overarching CCS manifest file) content are the Building Block Suppliers and the Integrator (as described in the TCCS Configuration - High Level Concept document), accordingly the CCS Configuration content (and the related Configuration Management process) is restricted to what these roles are capable to provide. The Operator will only enrich the CCS Configuration with definitions in terms of point in time and other operationally significant parameters for the activation.", -- "The CCS Configuration is an exhaustive, unambiguous description (all configuration items and an overarching CCS manifest file) of all field elements necessary to operate a physical instance of a CCS System, approved (in line with the CENELEC process) for deployment to a specific CCS Instance (identified via an identifier, for instance the already existing NID\_ENGINE, or other identifier already in use) and authorised to either operate on specific Networks (identified via unique Network Identifiers) or to be part of a specific Network.\n\nThe CCS Configuration is used for distribution of statical or semi-statical data (update data packet). It does not cover dynamic operational data like Moving Authority (for ETCS) or Journey Profiles (for ATO). The CCS Configuration is intended for CCS field components being present in multiple units, either close to infrastructure elements (e.g. CCS trackside instance and related signalling components like object controller) or CCS components on-board a train. It is not a prerequisite that CCS field components need to stay in operation when activating / installing a CCS Configuration (however, this is allowed). These components could shut down / restart, basically go out of operation, or remain in operation.\n\nThe main sources for the CCS Configuration (configuration items and overarching CCS manifest file) content are the Building Block Suppliers and the Integrator (as described in the TCCS Configuration - High Level Concept document), accordingly the CCS Configuration content (and the related Configuration Management process) is restricted to what these roles are capable to provide. The Operator will only enrich the CCS Configuration with definitions in terms of point in time and other operationally significant parameters for the activation."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CCS--FEATURE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ A CCS Feature is a main/top-level function on System Level 3 for CCS. Examples: ATO GoA2, Cold Movement Detection, C-DAS
\n
\n Mixable onboard features
\n(M/O means mandatory/optional)
\n
\nFeature(s) | M/O | Comment/Constrain/Condition/Version/Subfeature
\nETCS (ATP/ATO GoA1), only radio based | M | Compatibilty matrix onboard/trackside System Versions: Like today in TSI22. Assumption about phased out SV for the target system later on: Will be analysed in migration team. Inclusion of GSM-R in target architecture: To be analysed by MIG team with COM team, current TSI discussions to be included.
\nATO GoA 2,3 or 4 | O |
\nRTO | O |
\nAbsolute Continuous Train Positioning | O | Option: Capabal to use trackside augmentation information (like EGNOS or map information)
\nTrain Integrity Information | O | Mandatory, if no trackside train detection exists
\nReliable train length information | O | Mandatory, if no trackside train detection exists
\nCold movement detection | O |
\nDAC | O |
\nC-DAS | O |
\nStandard diagnostic interface | O |
\nTMS onboard user interface | O |
\nFallback/light ATP system | O | To be discussed. Idependent and simplified mode or system that has minimal dependecies/maximized availability and allows safe rudimentory traffic in most degraded modes or for special vehicles
\n Mixable trackside features in scope
\n(M/O means mandatory/optional)
\n
\n
\nFeature(s) | M/O | Comment/Constrain/Condition/Version/Subfeature
\nETCS onboard (ATP/ATO GoA1) | M | Radio based ETCS without lineside signals (no support for "overlay" installations.
\nATO GoA 2,3 or 4 | O |
\nRTO | O |
\nPostioning Augmentation Information | O | EGNOS, onboard map service
\nTrackside Train detection | O | Mandatory, if trains/train units have no train-integrity, position reporting and reliable length information. Could be block sensors, geometric positions or point sensors. Onboard/trackside sensors can exist in parallel.
\nDAC Control Applications | O |
\nC-DAS Service | O |
\nTMS services for onboard | O | e.g. request shunting route or possession by driver
\nSupport for fallback/light ATP system | O | To be discussed. Idependent and simplified mode or system that has minimal dependecies/maximized availability and allows safe rudimentory traffic in most degraded modes or for special vehicles """, -- """ A CCS Feature is a main/top-level function on System Level 3 for CCS. Examples: ATO GoA2, Cold Movement Detection, C-DAS\n\n\n\n Mixable onboard features\n\n(M/O means mandatory/optional)\n\n\n\nFeature(s) | M/O | Comment/Constrain/Condition/Version/Subfeature\n\nETCS (ATP/ATO GoA1), only radio based | M | Compatibilty matrix onboard/trackside System Versions: Like today in TSI22. Assumption about phased out SV for the target system later on: Will be analysed in migration team. Inclusion of GSM-R in target architecture: To be analysed by MIG team with COM team, current TSI discussions to be included.\n\nATO GoA 2,3 or 4 | O | \n\nRTO | O | \n\nAbsolute Continuous Train Positioning | O | Option: Capabal to use trackside augmentation information (like EGNOS or map information)\n\nTrain Integrity Information | O | Mandatory, if no trackside train detection exists\n\nReliable train length information | O | Mandatory, if no trackside train detection exists\n\nCold movement detection | O | \n\nDAC | O | \n\nC-DAS | O | \n\nStandard diagnostic interface | O | \n\nTMS onboard user interface | O | \n\nFallback/light ATP system | O | To be discussed. Idependent and simplified mode or system that has minimal dependecies/maximized availability and allows safe rudimentory traffic in most degraded modes or for special vehicles\n\n Mixable trackside features in scope\n\n(M/O means mandatory/optional)\n\n \n\n \n\nFeature(s) | M/O | Comment/Constrain/Condition/Version/Subfeature\n\nETCS onboard (ATP/ATO GoA1) | M | Radio based ETCS without lineside signals (no support for "overlay" installations.\n\nATO GoA 2,3 or 4 | O | \n\nRTO | O | \n\nPostioning Augmentation Information | O | EGNOS, onboard map service\n\nTrackside Train detection | O | Mandatory, if trains/train units have no train-integrity, position reporting and reliable length information. Could be block sensors, geometric positions or point sensors. Onboard/trackside sensors can exist in parallel.\n\nDAC Control Applications | O | \n\nC-DAS Service | O | \n\nTMS services for onboard | O | e.g. request shunting route or possession by driver\n\nSupport for fallback/light ATP system | O | To be discussed. Idependent and simplified mode or system that has minimal dependecies/maximized availability and allows safe rudimentory traffic in most degraded modes or for special vehicles """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CCS\_TMS--DATA--MODEL\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The CCS/TMS Data Model{comment:91} defines the harmonised language to generate and transport the Domain Data at System Pillar interfaces. The Transversal CCS Subdomain 1 (SD1) is responsible for the specification of the CCS/TMS Data Model in collaboration with
\n
\n\* the System Pillar domains which apply the defined data structures in interface specifications
\n\* the Innovation Pillar which proves the applicability of the data model by demonstrators.", -- "The CCS/TMS Data Model{comment:91} defines the harmonised language to generate and transport the Domain Data at System Pillar interfaces. The Transversal CCS Subdomain 1 (SD1) is responsible for the specification of the CCS/TMS Data Model in collaboration with\n\n \n\n\* the System Pillar domains which apply the defined data structures in interface specifications\n\n\* the Innovation Pillar which proves the applicability of the data model by demonstrators."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CMS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "CMS
\n Central modelling services", -- "CMS\n\n Central modelling services"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CM\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capacity Management.
\nIn the railway context, capacity management refers to the strategic planning and operational control measures implemented to effectively manage the available resources and infrastructure of a railway system in order to meet the demand for transportation services. It involves assessing, optimizing, and maximizing the capacity of various components within the railway system, such as tracks, stations, trains, and associated facilities.", -- "Capacity Management.\n\nIn the railway context, capacity management refers to the strategic planning and operational control measures implemented to effectively manage the available resources and infrastructure of a railway system in order to meet the demand for transportation services. It involves assessing, optimizing, and maximizing the capacity of various components within the railway system, such as tracks, stations, trains, and associated facilities."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:COMPONENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "{comment:35}
\nA component is used to structure functionality within building blocks. It implements and encapsulates one or more behaviors (business logic) and exposes services via defined interfaces to other components.{comment:159}", -- "{comment:35}\n\nA component is used to structure functionality within building blocks. It implements and encapsulates one or more behaviors (business logic) and exposes services via defined interfaces to other components.{comment:159}"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CONCEPTUAL--GLOSSARY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Container of concepts, typically class model illustrating a lexical definition, incl. relationships between concepts. May also be imported or authored by the project.
\nConceptual glossaries are referred to by lexcial glossaries.", -- "Container of concepts, typically class model illustrating a lexical definition, incl. relationships between concepts. May also be imported or authored by the project.\n\nConceptual glossaries are referred to by lexcial glossaries."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CONDITION--MONITORING--%3COF--AN--ITEM%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "obtaining information about physical state or operational parameters
\n Note 1 to entry: Condition monitoring is used to determine when preventive maintenance may be required.
\n Note 2 to entry: Condition monitoring may be conducted automatically during operation or at planned intervals.
\n Note 3 to entry: Condition monitoring methods include: vibration analysis, tribology and thermography.
\n [SOURCE: IEC 60050-192:2015, 192-06-28]", -- "obtaining information about physical state or operational parameters\n\n Note 1 to entry: Condition monitoring is used to determine when preventive maintenance may be required.\n\n Note 2 to entry: Condition monitoring may be conducted automatically during operation or at planned intervals.\n\n Note 3 to entry: Condition monitoring methods include: vibration analysis, tribology and thermography.\n\n [SOURCE: IEC 60050-192:2015, 192-06-28]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CONFIDENTIALITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Confidentiality, in the context of computer systems, allows only authorised users to access protected data using specific mechanisms to ensure confidentiality and safeguard data from harmful intrusion.
\n (source: SPPRAMSS-1705 - [UNSIG Subset-146] )", -- "Confidentiality, in the context of computer systems, allows only authorised users to access protected data using specific mechanisms to ensure confidentiality and safeguard data from harmful intrusion.\n\n (source: SPPRAMSS-1705 - [UNSIG Subset-146] )"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CONFIGURATION--ITEM\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "part of the system that must be enginereed, produced or bought, duplicated as much as it is used in the system and
\n assembled with others in order to build a copy of the system", -- "part of the system that must be enginereed, produced or bought, duplicated as much as it is used in the system and \n\n assembled with others in order to build a copy of the system"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CONFIGURATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Description of a collection of model elements, either available or unavailable in a given context. A context can be a mode, a
\nstate or a combination of both.", -- "Description of a collection of model elements, either available or unavailable in a given context. A context can be a mode, a \n\nstate or a combination of both."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CONSOLIDATED--GLOSSARY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A SPPR-5436 - Lexical Glossary is called consolidated if all contained items are not contradictory nor ambiguous toward other glossaries of System Pillar. All possible ambiguities are resolved by means of context information added to the definitions.
\nA consolidated glossary provides a safe basis for the context underlying its namespace. This supports efficient communication in expert groups.
\nSee SPPR-5313 - Consolidate the glossary items.", -- "A SPPR-5436 - Lexical Glossary is called consolidated if all contained items are not contradictory nor ambiguous toward other glossaries of System Pillar. All possible ambiguities are resolved by means of context information added to the definitions. \n\nA consolidated glossary provides a safe basis for the context underlying its namespace. This supports efficient communication in expert groups. \n\nSee SPPR-5313 - Consolidate the glossary items."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:CORRECTIVE--MAINTENANCE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Maintenance carried out after fault detection to effect restoration
\n
\nNote 1 to entry: Corrective maintenance of software invariably involves some modification.
\n [SOURCE: IEC 60050-192:2015, 192-06-06]", -- "Maintenance carried out after fault detection to effect restoration \n\n \n\nNote 1 to entry: Corrective maintenance of software invariably involves some modification.\n\n [SOURCE: IEC 60050-192:2015, 192-06-06]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:DATA--MODEL--LAYER\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Note: informelle Gruppierung im Modell (analog zur RCA tiers), für Kommunikation/Verständnis
\nA data model layer offers a view of the information as needed by, and as limited by a functional domain such as topology, geometry, signalling, gauge, etc.", -- "Note: informelle Gruppierung im Modell (analog zur RCA tiers), für Kommunikation/Verständnis\n\nA data model layer offers a view of the information as needed by, and as limited by a functional domain such as topology, geometry, signalling, gauge, etc."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:DEFINITION\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "ARCADIA (Architecture Analysis & Design Integrated Approach) is a system and software architecture engineering method based on architecture-centric and model-driven engineering activities. Main resources about ARCADIA can be found on the official websites: https://www.eclipse.org/capella/arcadia-reference.html
\n
\n (image: 1-screenshot-20230107-163323.png)", -- "ARCADIA (Architecture Analysis & Design Integrated Approach) is a system and software architecture engineering method based on architecture-centric and model-driven engineering activities. Main resources about ARCADIA can be found on the official websites: https://www.eclipse.org/capella/arcadia-reference.html \n\n \n\n (image: 1-screenshot-20230107-163323.png)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:DEVELOPMENT--TASK\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Take spec, build system
\n\* Usually starts on CENELEC P5
\n\* Bottom of CENELEC “V”", -- "\* Take spec, build system\n\n\* Usually starts on CENELEC P5\n\n\* Bottom of CENELEC “V”"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:DEVICE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Device{comment:36}
\n A physical entity performing a predefined (set of) task(s). It consists of software integrated on a hardware.", -- "Device{comment:36}\n\n A physical entity performing a predefined (set of) task(s). It consists of software integrated on a hardware."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:DISTRIBUTION--JOB\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Distribution Job is the complete configuration data packet that is finally provided to the building blocks. The Distribution Job contains the metadata required for the distribution of a CCS Configuration. In addition to linking a referenced CCS Manifest to a CCS Vehicle Identifier or a CCS Trackside Identifier, it also includes information like distribution date / time, activation location, etc. This metadata will be evaluated by central location system (centrally located: typically a CCS configuration management system) and different distributed systems (distributed in the field: e.g. on-board) involved in the distribution process.", -- "Distribution Job is the complete configuration data packet that is finally provided to the field components. The Distribution Job contains the metadata required for the distribution of a CCS Configuration. In addition to linking a referenced CCS Manifest to a CCS Vehicle Identifier or a CCS Trackside Identifier, it also includes information like distribution date / time, activation location, etc. This metadata will be evaluated by central location system (centrally located: typically a CCS configuration management system) and different distributed systems (distributed in the field: e.g. on-board) involved in the distribution process."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ERTMS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "European Railway Traffic Management System (ERTMS) is a single European signalling and speed control system that ensures interoperability of the national railway systems, reducing the purchasing and maintenance costs of the signalling systems as well as increasing the speed of trains, the capacity of infrastructure and the level of safety in rail transport. ERTMS comprises of the European Train Control System (ETCS), i.e. a cab-signalling system that incorporates automatic train protection, the Global System for Mobile communications for Railways (GSM-R) and operating rules. (ERA definition)", -- "ERTMS comprises of the European Train Control System (ETCS), i.e. a cab-signalling system that incorporates automatic train protection, the Global System for Mobile communications for Railways (GSM-R) and operating rules.\n(ERA definition)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ERTMS\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "European Rail Traffic Management System", -- "European Railway Traffic Management System (ERTMS) is a single European signalling and speed control system that ensures interoperability of the national railway systems, reducing the purchasing and maintenance costs of the signalling systems as well as increasing the speed of trains, the capacity of infrastructure and the level of safety in rail transport. ERTMS comprises of the European Train Control System (ETCS), i.e. a cab-signalling system that incorporates automatic train protection, the Global System for Mobile communications for Railways (GSM-R) and operating rules. (ERA definition)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ETCS\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "European Train Control System (ETCS) is a cab-signalling system that incorporates automatic train protection (ERA definition).
\nETCS in the frame of SERA will support the following Levels:
\n\* L2 with fixed train detection (classic L2 with trackside detection (track circuits, axle-counters, …))
\n\* L2 with hybrid train detection (virtual fixed or moving blocks with trackside detection (axle-counters, …)) – formerly known as Hybrid L3
\n\* L2 with virtual fixed or moving blocks and train integrity (no trackside detection) – formerly known as pure L3", -- "European Train Control System (ETCS) is a cab-signalling system that incorporates automatic train protection (ERA definition).\n\nETCS in the frame of SERA will support the following Levels: \n\n\* L2 with fixed train detection (classic L2 with trackside detection (track circuits, axle-counters, …))\n\n\* L2 with hybrid train detection (virtual fixed or moving blocks with trackside detection (axle-counters, …)) – formerly known as Hybrid L3\n\n\* L2 with virtual fixed or moving blocks and train integrity (no trackside detection) – formerly known as pure L3"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:EU-RAILGOVERNING--BOARD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "EU-RailGoverning Board
\n Final decision body, where a decisions are adopted a majority of at least 55% of the votes", -- "EU-RailGoverning Board\n\n Final decision body, where a decisions are adopted a majority of at least 55% of the votes"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:EVOLVABILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Evolvability is the ability to easily adapt to new technologies or to extend the functionality of the CCS system without the involvement of the original supplier.{comment:429}", -- "Evolvability is the ability to easily adapt to new technologies or to extend the functionality of the CCS system without the involvement of the original supplier."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:EXCHANGEABILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Exchangeability is the ability to replace a sub-system from supplier A by a subsystem from supplier B without affecting other sub-systems or the overall system/subsystem and with a reasonable integration effort and/or certification effort. Exchangeability and interchangeability are related to the physical characteristics of sub-systems whereas interoperability is related to interactions between sub-systems (e.g. also between STM and ETCS on-board there is interoperability).", -- "Exchangeability is the ability to replace a sub-system from supplier A by a subsystem from supplier B without affecting other sub-systems or the overall system/subsystem and with a reasonable integration effort and/or certification effort. Exchangeability and interchangeability are related to the physical characteristics of sub-systems wheres interoperability is related to interactions between subsystems (e.g. also between STM and ETCS on-board there is interoperability)."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAIL-SAFE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "able to enter or remain in a safe state in the event of a failure
\n[SOURCE: IEC 60050-821:2017 , 821-01-10]", -- "able to enter or remain in a safe state in the event of a failure\n\n[SOURCE: IEC 60050-821:2017 , 821-01-10]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAIL-SAFE\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "able to enter or remain in a safe state in the event of a failure
\n[821-01-10 ]", -- "able to enter or remain in a safe state in the event of a failure\n\n[821-01-10 ]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAILURE--CAUSE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "set of circumstances that leads to failure
\nNote 1 to entry: A failure cause may originate during specification, design, manufacture, installation, operation or maintenance of an item.
\n[SOURCE: IEC 60050.192:2015, 192-03-11]", -- "set of circumstances that leads to failure\n\nNote 1 to entry: A failure cause may originate during specification, design, manufacture, installation, operation or maintenance of an item.\n\n[SOURCE: IEC 60050.192:2015, 192-03-11]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAILURE--MODE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Manner in which failure occurs.
\n Note 1 to entry: A failure mode may be defined by the function lost or other state transition that occurred.
\n[SOURCE: IEC 60050-192:2015, 192-03-17]", -- "Manner in which failure occurs.\n\n Note 1 to entry: A failure mode may be defined by the function lost or other state transition that occurred.\n\n[SOURCE: IEC 60050-192:2015, 192-03-17]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAILURE--RATE--\_821-12-21\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "limit of the ratio of the conditional probability that the instant of time, T, of a failure of a product falls within a given time interval (t, t + Δt) and the duration of this interval, Δt, when Δt tends towards zero, given that the item is in an up state at the start of the time interval
\n Note 1 to entry: For applications where distance travelled or number of cycles of operation is more relevant than time then the unit of time can be replaced by the unit of distance or cycles, as appropriate.
\n
\n Note 2 to entry: The term “failure rate” is often used in the sense of “mean failure rate” defined in IEV 192-05-07.
\n [SOURCE: IEC 62278:2002, 3.14, modified]", -- "limit of the ratio of the conditional probability that the instant of time, T, of a failure of a product falls within a given time interval (t, t + Δt) and the duration of this interval, Δt, when Δt tends towards zero, given that the item is in an up state at the start of the time interval\n\n Note 1 to entry: For applications where distance travelled or number of cycles of operation is more relevant than time then the unit of time can be replaced by the unit of distance or cycles, as appropriate.\n\n\n\n Note 2 to entry: The term “failure rate” is often used in the sense of “mean failure rate” defined in IEV 192-05-07.\n\n [SOURCE: IEC 62278:2002, 3.14, modified]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAILURE--RATE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "limit of the ratio of the conditional probability that the instant of time, T, of a failure of a product falls within a given time interval (t, t + Δt) and the duration of this interval, Δt, when Δt tends towards zero, given that the item is in an up state at the start of the time interval
\n Note 1 to entry: For applications where distance travelled or number of cycles of operation is more relevant than time then the unit of time can be replaced by the unit of distance or cycles, as appropriate.
\n
\n Note 2 to entry: The term “failure rate” is often used in the sense of “mean failure rate” defined in IEV 192-05-07.
\n [IEC 60050-821, 821-12-21]", -- "limit of the ratio of the conditional probability that the instant of time, T, of a failure of a product falls within a given time interval (t, t + Δt) and the duration of this interval, Δt, when Δt tends towards zero, given that the item is in an up state at the start of the time interval\n\n Note 1 to entry: For applications where distance travelled or number of cycles of operation is more relevant than time then the unit of time can be replaced by the unit of distance or cycles, as appropriate.\n\n\n\n Note 2 to entry: The term “failure rate” is often used in the sense of “mean failure rate” defined in IEV 192-05-07.\n\n [821-12-21][IEC 62278:2002, 3.14, modified]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAILURE--RATE\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "limit of the ratio of the conditional probability that the instant of time, T, of a failure of a product falls within a given time interval (t, t + Δt) and the duration of this interval, Δt, when Δt tends towards zero, given that the item is in an up state at the start of the time interval
\n Note 1 to entry: For applications where distance travelled or number of cycles of operation is more relevant than time then the unit of time can be replaced by the unit of distance or cycles, as appropriate.
\n
\n Note 2 to entry: The term “failure rate” is often used in the sense of “mean failure rate” defined in IEV 192-05-07.
\n [821-12-21][IEC 62278:2002, 3.14, modified]", -- "limit of the ratio of the conditional probability that the instant of time, T, of a failure of a product falls within a given time interval (t, t + Δt) and the duration of this interval, Δt, when Δt tends towards zero, given that the item is in an up state at the start of the time interval\n\n Note 1 to entry: For applications where distance travelled or number of cycles of operation is more relevant than time then the unit of time can be replaced by the unit of distance or cycles, as appropriate.\n\n\n\n Note 2 to entry: The term “failure rate” is often used in the sense of “mean failure rate” defined in IEV 192-05-07.\n\n [821-12-21][IEC 62278:2002, 3.14, modified]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAILURE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ loss of ability to perform as required
\n Note 1 to entry: A failure of an item is an event that results in a fault of that item: see "fault" (IEV 192-04-01).
\nNote 2 to entry: Qualifiers, such as catastrophic, critical, major, minor, marginal and insignificant, can be used to categorize failures according to the severity of consequences, the choice and definitions of severity criteria depending upon the field of application.
\nNote 3 to entry: Qualifiers, such as misuse, mishandling and weakness, can be used to categorize failures according to the cause of failure.
\n [SOURCE: IEC 60050-192:2015, 192-03-01] """, -- """ loss of ability to perform as required \n\n Note 1 to entry: A failure of an item is an event that results in a fault of that item: see "fault" (IEV 192-04-01).\n\nNote 2 to entry: Qualifiers, such as catastrophic, critical, major, minor, marginal and insignificant, can be used to categorize failures according to the severity of consequences, the choice and definitions of severity criteria depending upon the field of application.\n\nNote 3 to entry: Qualifiers, such as misuse, mishandling and weakness, can be used to categorize failures according to the cause of failure.\n\n [SOURCE: IEC 60050-192:2015, 192-03-01] """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAULT--%3COF--AN--ITEM%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "inability to perform as required, due to an internal state
\nNote 1 to entry: A fault of an item results from a failure, either of the item itself, or from a deficiency in an earlier stage of the life cycle, such as specification, design, manufacture or maintenance. See latent fault (192-04-08).{comment:215}
\nNote 2 to entry: Qualifiers, such as specification, design, manufacture, maintenance or misuse, may be used to indicate the cause of a fault.
\nNote 3 to entry: The type of fault may be associated with the type of associated failure, e.g. wear-out fault and wear-out failure.
\nNote 4 to entry: The adjective “faulty” designates an item having one or more faults.
\n[SOURCE: IEC 60050-192:2015, 192-04-01]", -- "inability to perform as required, due to an internal state\n\nNote 1 to entry: A fault of an item results from a failure, either of the item itself, or from a deficiency in an earlier stage of the life cycle, such as specification, design, manufacture or maintenance. See latent fault (192-04-08).\n\nNote 2 to entry: Qualifiers, such as specification, design, manufacture, maintenance or misuse, may be used to indicate the cause of a fault.\n\nNote 3 to entry: The type of fault may be associated with the type of associated failure, e.g. wear-out fault and wear-out failure.\n\nNote 4 to entry: The adjective “faulty” designates an item having one or more faults.\n\n[SOURCE: IEC 60050-192:2015, 192-04-01]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAULT--DETECTION--TIME\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "DEPRECATED: undetected fault time
\ntime interval between failure and detection of the resulting fault
\n[SOURCE: IEC 60050-192:2015,192-07-11]", -- "DEPRECATED: undetected fault time \n\ntime interval between failure and detection of the resulting fault \n\n[SOURCE: IEC 60050-192:2015,192-07-11]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FAULT--TREE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "logic diagram showing the faults of sub items, external events, or combinations thereof, which cause a predefined, undesired event
\n[SOURCE: IEC 60050-192:2015, 192-11-07]", -- "logic diagram showing the faults of sub items, external events, or combinations thereof, which cause a predefined, undesired event\n\n[SOURCE: IEC 60050-192:2015, 192-11-07]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FEATURE--SUMMARY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The ARCADIA method:
\n\* Covers all structured engineering activities, from capturing customer operational needs to system integration verification validation (IVV);
\n\* Takes into account multiple engineering levels and their effective collaboration (system, subsystem, software, hardware, etc.);
\n\* Integrates co-engineering with specialty engineering (safety, security, performance, interfaces, logistics ...) and IVV;
\n\* Provides the ability not only to share descriptive models but also to collaboratively validate properties of the definition and the architecture;
\n\* Is field-tested in full-scale industrial applications, and is currently deployed on dozens of major projects in several countries and divisions of Thales.", -- "The ARCADIA method: \n\n\* Covers all structured engineering activities, from capturing customer operational needs to system integration verification validation (IVV);\n\n\* Takes into account multiple engineering levels and their effective collaboration (system, subsystem, software, hardware, etc.);\n\n\* Integrates co-engineering with specialty engineering (safety, security, performance, interfaces, logistics ...) and IVV;\n\n\* Provides the ability not only to share descriptive models but also to collaboratively validate properties of the definition and the architecture;\n\n\* Is field-tested in full-scale industrial applications, and is currently deployed on dozens of major projects in several countries and divisions of Thales."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FFFIS-----FORM--FIT--FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A FFFIS is the complete definition of an interface between functional or physical entities.
\n The FFFIS includes:
\n - FIS,
\n - Electrical characteristics related to data,
\n - communication protocol,
\n - and including connector and plug.
\n The FFFIS and accompanying documents (e.g. safety analysis) guarantees the interoperability but not the exchangeability of physical entities, see Subset-037 SPT2ARC-1620.", -- "A FFFIS is the complete definition of an interface between functional or physical entities.\n\n The FFFIS includes:\n\n - FIS,\n\n - Electrical characteristics related to data,\n\n - communication protocol,\n\n - and including connector and plug.\n\n he FFFIS and accompanying documents (e.g. safety analysis) guarantees the interoperability but not the exchangeability of physical entities. [Subset-037]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FIS-----FUNCTIONAL--INTERFACE--SPECIFICATION\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "FIS - Functional Interface Specification", -- "Functional requirements and description of an interface. This clarifies also the functional apportionment and related safety requirements of the two sub-systems of the interface."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FMECA\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ failure modes, effects and criticality analysis
\nquantitative or qualitative method of analysis that involves failure modes and effects analysis together with a consideration of the probability of the failure mode occurrence and the severity of the effects
\nNote 1 to entry: The term "fault mode, effects and criticality analysis" in IEC 60050-191:1990 (now withdrawn; replaced by IEC 60050-192:2015) is deprecated, since a fault (192-04-01) is a state and cannot logically have a mode, whereas a failure mode (192-03-17) is a change of state.
\n[SOURCE: IEC 60050-192:2015, 192-11-06]
\n
\n Note 2 to entry: FMEA is a systematic method of evaluating an item or process to identify the ways in which it might potentially fail, and the effects of the mode of failure upon the performance of the item or process and on the surrounding environment and personnel.
\nFailure modes may be prioritized according to their importance. The prioritization can be based on a ranking of the severity alone, or this can be combined with other measures of importance. When failure modes are prioritized, the process is referred to as failure modes, effects and criticality analysis (FMECA). """, -- """ failure modes, effects and criticality analysis\n\nquantitative or qualitative method of analysis that involves failure modes and effects analysis together with a consideration of the probability of the failure mode occurrence and the severity of the effects\n\nNote 1 to entry: The term "fault mode, effects and criticality analysis" in IEC 60050-191:1990 (now withdrawn; replaced by IEC 60050-192:2015) is deprecated, since a fault (192-04-01) is a state and cannot logically have a mode, whereas a failure mode (192-03-17) is a change of state.\n\n[SOURCE: IEC 60050-192:2015, 192-11-06]\n\n\n\n Note 2 to entry: FMEA is a systematic method of evaluating an item or process to identify the ways in which it might potentially fail, and the effects of the mode of failure upon the performance of the item or process and on the surrounding environment and personnel.\n\nFailure modes may be prioritized according to their importance. The prioritization can be based on a ranking of the severity alone, or this can be combined with other measures of importance. When failure modes are prioritized, the process is referred to as failure modes, effects and criticality analysis (FMECA). """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FMECA\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Failure modes and effects analysis (FMEA) is a systematic method of evaluating an item or process to identify the ways in which it might potentially fail, and the effects of the mode of failure upon the performance of the item or process and on the surrounding environment and personnel.
\nFailure modes may be prioritized according to their importance. The prioritization can be based on a ranking of the severity alone, or this can be combined with other measures of importance. When failure modes are prioritized, the process is referred to as failure modes, effects and criticality analysis (FMECA).", -- "Failure modes and effects analysis (FMEA) is a systematic method of evaluating an item or process to identify the ways in which it might potentially fail, and the effects of the mode of failure upon the performance of the item or process and on the surrounding environment and personnel.\n\nFailure modes may be prioritized according to their importance. The prioritization can be based on a ranking of the severity alone, or this can be combined with other measures of importance. When failure modes are prioritized, the process is referred to as failure modes, effects and criticality analysis (FMECA)."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FOLLOW--A--TRACE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A trace (nodes/work items and their links) is a graph that works out one issue (e.g. one requirement) in the most efficient, complete and fast way, down to its operational and system implementation
\n (image: 2-image2.png)
\nFigure {caption:Figure}: Trace for a CBO requirement (red point in the middle) in an ALM System
\n
\nSystem Levels are just “areas of team responsibilities” for working on parts of the graph
\n\* System Level 1 / 2 = teams … = Work item detail level …
\n\* System Level 3 = teams … = Work item detail level …
\n\* System Level 4 = teams … = Work item detail level …
\n\* System Level 5 = teams … = Work item detail level …
\nAssigning a work item to a System Level just means: Assigning them to a team.
\n No good idea: Design every System Level with everything that a real-world system would need. Systems on Level 1-4 are no real-world systems, they just structure the work assignment.
\n Trace Example – just one “branch”: The reality is not puristic, it is like a “neuron-mesh”
\nA. SL2: Requirement “High, scalable, and flexible transport capacity” >>
\nB. SL3: Process “efficient ATP for high density lines” >>
\nC. SL4: Requirement “Precise and frequent localisation” in the train >>
\nD. SL3: Architecture (functions) for a high- performance localisation System >>
\nE. SL5: Interface requirements to deliver a map to the train >>
\nF. SL5: Requirement that Traffic CS delivers a reliable map >> ….
\n
\n Think in “good complete traces”, and do not “fill all System Levels” (!)
\n--> It is just important that “traces are complete, good and correct”,
\n with work items assigned to the right team level
\n--> What System Levels are NOT for: Being the basic scheme for everything
\n\* Creating a break-down element on every System Level --> this does not work and requires too much effort
\n\* Exception: The work item “system” is broken down on every level, but for example interfaces or logical components only exist in System Level 5)
\nWhen do we need “completion” on System Levels? Sometimes it is necessary.
\n Example 1: Safety analysis for “Assure that all CCS processes are safe”
\n\* All operational processes need to be listed and analysed on System level 3 (OA)
\n\* Out of this all traces have to be analysed to understand the hazards and risks
\nBut this does not mean that this is also done for System Level 1,2, or 4.
\n Just the operational design team in Task 2 has to do this.
\n Example 2: Systems on System Level 5 (“standard products”)
\n\* Complete description how to install and use them (processes)
\n\* Complete functional description
\n\* Complete system and interface description
\nHow to follow “traces”
\nFollowing a trace means to break down work items more and more. Every breakdown is not necessarily “homogenous”. One breakdown step (indicative):
\n
\n (image: 1-screenshot-20221229-205247.png)
\n
\nA break down step follows this workflow (for the assigned team)
\n\* Assess if work item is accepted and makes sense. If not, reject and forward work item to a functional team
\n\* Analyse, work out, and refine a work item; change status when finished
\n\* Draft and link the derived work items, set status to “proposed”
\n\* Propose their assignment to a team
\n\* Assigned team accepts or forwards the derived work item to a functional team.
\nHow do traces start, how to reach “completeness”?
\nA. New traces start from
\n\* Any input of the stakeholders
\n\* Common Business Objectives (CBO)
\n\* A System Level 2/3 mission and the needed operational capabilities, derived operational scenarios and their operational requirements
\n\* Any input inside of the System Pillar team – if a backward linking to CBO or operational missions/capabilities is possible or stakeholder (steering group) agree
\nCompleteness is reached if for all of these (decided) inputs a trace down to the implementation exists with a valid trace.
\nB. A trace is valid if
\n\* …it is connected up to a decided demand (A.)
\n\* …it is connected down to an implementation in operational processes and System Level 5.
\n\* … all links between work items of the trace are valid (correct derivation)
\nThe status of all traces will be visible at any time via the ALM.", -- "A trace (nodes/work items and their links) is a graph that works out one issue (e.g. one requirement) in the most efficient, complete and fast way, down to its operational and system implementation\n\n (image: 2-image2.png) \n\nFigure {caption:Figure}: Trace for a CBO requirement (red point in the middle) in an ALM System\n\n \n\nSystem Levels are just “areas of team responsibilities” for working on parts of the graph \n\n\* System Level 1 / 2 = teams … = Work item detail level …\n\n\* System Level 3 = teams … = Work item detail level …\n\n\* System Level 4 = teams … = Work item detail level …\n\n\* System Level 5 = teams … = Work item detail level …\n\nAssigning a work item to a System Level just means: Assigning them to a team.\n\n No good idea: Design every System Level with everything that a real-world system would need. Systems on Level 1-4 are no real-world systems, they just structure the work assignment.\n\n Trace Example – just one “branch”: The reality is not puristic, it is like a “neuron-mesh”\n\nA. SL2: Requirement “High, scalable, and flexible transport capacity” >>\n\nB. SL3: Process “efficient ATP for high density lines” >>\n\nC. SL4: Requirement “Precise and frequent localisation” in the train >>\n\nD. SL3: Architecture (functions) for a high- performance localisation System >>\n\nE. SL5: Interface requirements to deliver a map to the train >>\n\nF. SL5: Requirement that Traffic CS delivers a reliable map >> ….\n\n \n\n Think in “good complete traces”, and do not “fill all System Levels” (!)\n\n--> It is just important that “traces are complete, good and correct”, \n\n with work items assigned to the right team level\n\n--> What System Levels are NOT for: Being the basic scheme for everything \n\n\* Creating a break-down element on every System Level --> this does not work and requires too much effort\n\n\* Exception: The work item “system” is broken down on every level, but for example interfaces or logical components only exist in System Level 5)\n\nWhen do we need “completion” on System Levels? Sometimes it is necessary.\n\n Example 1: Safety analysis for “Assure that all CCS processes are safe” \n\n\* All operational processes need to be listed and analysed on System level 3 (OA)\n\n\* Out of this all traces have to be analysed to understand the hazards and risks\n\nBut this does not mean that this is also done for System Level 1,2, or 4.\n\n Just the operational design team in Task 2 has to do this.\n\n Example 2: Systems on System Level 5 (“standard products”) \n\n\* Complete description how to install and use them (processes)\n\n\* Complete functional description\n\n\* Complete system and interface description\n\nHow to follow “traces”\n\nFollowing a trace means to break down work items more and more. Every breakdown is not necessarily “homogenous”. One breakdown step (indicative):\n\n \n\n (image: 1-screenshot-20221229-205247.png) \n\n \n\nA break down step follows this workflow (for the assigned team) \n\n\* Assess if work item is accepted and makes sense. If not, reject and forward work item to a functional team\n\n\* Analyse, work out, and refine a work item; change status when finished\n\n\* Draft and link the derived work items, set status to “proposed”\n\n\* Propose their assignment to a team\n\n\* Assigned team accepts or forwards the derived work item to a functional team.\n\nHow do traces start, how to reach “completeness”?\n\nA. New traces start from \n\n\* Any input of the stakeholders\n\n\* Common Business Objectives (CBO)\n\n\* A System Level 2/3 mission and the needed operational capabilities, derived operational scenarios and their operational requirements\n\n\* Any input inside of the System Pillar team – if a backward linking to CBO or operational missions/capabilities is possible or stakeholder (steering group) agree\n\nCompleteness is reached if for all of these (decided) inputs a trace down to the implementation exists with a valid trace.\n\nB. A trace is valid if \n\n\* …it is connected up to a decided demand (A.)\n\n\* …it is connected down to an implementation in operational processes and System Level 5.\n\n\* … all links between work items of the trace are valid (correct derivation)\n\nThe status of all traces will be visible at any time via the ALM."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FTA\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "fault tree analysis
\ndeductive analysis using fault trees
\nNote 1 to entry: See also fault tree (192-11-07 SPPRAMSS-4464 - fault tree ).
\n[SOURCE: IEC 60050-192:2015, 192-11-08]", -- "fault tree analysis\n\ndeductive analysis using fault trees\n\nNote 1 to entry: See also fault tree (192-11-07 SPPRAMSS-4464 - fault tree ).\n\n[SOURCE: IEC 60050-192:2015, 192-11-08]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FULL--BACKWARDS--COMPATIBILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ Full backward compatibility, also known as "complete backward compatibility," refers to a situation in which a newer version of software, hardware, or a system is not only compatible with the previous version but also ensures compatibility with all previous versions and features without exceptions. In other words, any software, data, or hardware that worked with older versions will work seamlessly and without any issues with the new version.
\n
\nWith full backward compatibility, users can transition to the latest version with confidence, knowing that they won't encounter any incompatibilities or disruptions.
\n This level of compatibility often requires careful design and testing to ensure that all legacy functionalities and components are supported and function correctly in the newer version. """, -- """ Full backward compatibility, also known as "complete backward compatibility," refers to a situation in which a newer version of software, hardware, or a system is not only compatible with the previous version but also ensures compatibility with all previous versions and features without exceptions. In other words, any software, data, or hardware that worked with older versions will work seamlessly and without any issues with the new version.\n\n\n\nWith full backward compatibility, users can transition to the latest version with confidence, knowing that they won't encounter any incompatibilities or disruptions.\n\n This level of compatibility often requires careful design and testing to ensure that all legacy functionalities and components are supported and function correctly in the newer version. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FUNCTIONAL--APPLICATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A comprehensive set of self-contained software functions, assumed to be provided as one product by a single vendor. A functional application could consist of:
\n
\n\* 1..n software functions and
\n\* a generic (deployment) configuration.
\nNote that runtime environment is not part of the functional application, even if it is sometimes (not always) exchanged within the same step as the application is changed.", -- "A comprehensive set of self-contained software functions, assumed to be provided as one product by a single vendor. A functional application could consist of:\n\n \n\n\* 1..n software functions and\n\n\* a generic (deployment) configuration.\n\nNote that runtime environment is not part of the functional application, even if it is sometimes (not always) exchanged within the same step as the application is changed."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FUNCTIONAL--APPORTIONMENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Functional apportionment is the clear assignment of functions to sub-systems. It is an architectural choice supporting the ability to replace a sub-system of supplier A by a subsystem of supplier B both compliant with a given FIS.", -- "Functional apportionment is an architectural choice supporting the ability to replace a sub-system of supplier A by a subsystem of supplier B both compliant with a given FIS."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FUNCTIONAL--TEAM\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ A Functional team includes the role owners of the same role. It organizes the work allocation and overall results for that role, including the workflows and working methods.
\n
\nThe name of the functional team has a “-F” at the end.
\n
\nThe functional teams form a matrix to the organisational units of the SP.
\n
\nIf an organisational unit has more persons with the same role, one of them can be delegated to the functional team as a "speaker".
\n
\nFunctional teams organise themselves internally. From "outside" (SP processes) the team is addressed as one actor. The worksplit in the functional teams is decided in the team. The functional teams act with the responsibilities and tasks defined by the role it represents. The functional team is lead and moderated by the participant from System Level 1/ (mostly people from the engineering services, see role allocation table).
\n
\nFunctional teams decide in consensus, or escalate to the coregroup.
\n
\nExample for the functional Team “REQ-F”
\n
\n (image: 1-Bild\_2.png)
\n (image: 2-Grafik\_1.png) """, -- """ A Functional team includes the role owners of the same role. It organizes the work allocation and overall results for that role, including the workflows and working methods. \n\n \n\nThe name of the functional team has a “-F” at the end.\n\n \n\nThe functional teams form a matrix to the organisational units of the SP. \n\n \n\nIf an organisational unit has more persons with the same role, one of them can be delegated to the functional team as a "speaker". \n\n \n\nFunctional teams organise themselves internally. From "outside" (SP processes) the team is addressed as one actor. The worksplit in the functional teams is decided in the team. The functional teams act with the responsibilities and tasks defined by the role it represents. The functional team is lead and moderated by the participant from System Level 1/ (mostly people from the engineering services, see role allocation table). \n\n \n\nFunctional teams decide in consensus, or escalate to the coregroup. \n\n \n\nExample for the functional Team “REQ-F”\n\n \n\n (image: 1-Bild\_2.png) \n\n (image: 2-Grafik\_1.png) """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FUNCTION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A function transforms several one or more input values to several one or more output values according to the function’s behaviour. The function’s behaviour is defined by one or multiple functional requirements, defining “what” the function is doing. Additional non-functional requirements attached to the function define the quality attributes of “how” (how safe, how accurate, how fast, how reliable, etc.) the function is performing the transformation.
\nSome non-functional requirements, e.g., on weight constraints or physical dimension constraints), will also be allocated to subsystems.", -- "A function transforms several one or more input values to several one or more output values according to the function’s behaviour. The function’s behaviour is defined by one or multiple functional requirements, defining “what” the function is doing. Additional non-functional requirements attached to the function define the quality attributes of “how” (how safe, how accurate, how fast, how reliable, etc.) the function is performing the transformation.\n\nSome non-functional requirements, e.g., on weight constraints or physical dimension constraints), will also be allocated to subsystems."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:FUNKTIONAL--TEAM\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ A Functional team includes the role owners of the same role. It organizes the work allocation and overall results for that role, including the workflows and working methods.
\nThe name of the functional team has a “-F” at the end.
\nThe functional teams form a matrix to the organisational units of the SP.
\nIf an organisational unit has more persons with the same role, one of them can be delegated to the functional team as a "speaker".
\nFunctional teams organise themselves internally. From "outside" (SP processes) the team is addressedaddressed as one actor. The worksplit in the functional teams is decided in the team. The functional teams act with the responsibilities and tasks defined by the role it represents. The functional team is lead and moderated by the participant from System Level 1/ (mostly people from the engineering services, see role allocation table).
\nFunctional teams decide in consensus, or escalate to the coregroup.
\nExample for the functional Team “REQ-F” """, -- """ A Functional team includes the role owners of the same role. It organizes the work allocation and overall results for that role, including the workflows and working methods. \n\nThe name of the functional team has a “-F” at the end.\n\nThe functional teams form a matrix to the organisational units of the SP. \n\nIf an organisational unit has more persons with the same role, one of them can be delegated to the functional team as a "speaker". \n\nFunctional teams organise themselves internally. From "outside" (SP processes) the team is addressedaddressed as one actor. The worksplit in the functional teams is decided in the team. The functional teams act with the responsibilities and tasks defined by the role it represents. The functional team is lead and moderated by the participant from System Level 1/ (mostly people from the engineering services, see role allocation table). \n\nFunctional teams decide in consensus, or escalate to the coregroup. \n\nExample for the functional Team “REQ-F” """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:GENERIC--WORKFLOW--TYPES\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Assignment workflow: Unassigned, rejected, or unlinked work items are assigned to a team based on the work item type and system level (done by a functional team, see [2])
\n\* Derivation workflow: The standard workflow, where work items are resolved along the mandatory trace (along the workflow rules), work is moving from team to team, or from team collaboration to team collaboration
\n\* Trace change workflow: A change in a trace is analysed up to the highest work item again (done by modelling service, all work items get “suspect”), and from there all derivations are checked, resolved, and approved again (derivation workflow).
\n\* Uplink workflow: A set of traces is “connected” to another set. This typically happens, when “external contributions” are imported (e.g. a team that proposes the interfaces of a subsystem). The connection links are created as proposals. For all new connection links a trace change workflow is executed. Redundancies are analysed and eliminated, what again triggers trace change workflows.
\n\* Trace analysis workflow: Progress and trace consistency are analysed and reported by the modelling service
\nThese workflows are assigned by the team or person, where the triggering work item is assigned to. The trace analysis is done by the modelling service on a frequent basis.", -- "\* Assignment workflow: Unassigned, rejected, or unlinked work items are assigned to a team based on the work item type and system level (done by a functional team, see [2])\n\n\* Derivation workflow: The standard workflow, where work items are resolved along the mandatory trace (along the workflow rules), work is moving from team to team, or from team collaboration to team collaboration\n\n\* Trace change workflow: A change in a trace is analysed up to the highest work item again (done by modelling service, all work items get “suspect”), and from there all derivations are checked, resolved, and approved again (derivation workflow).\n\n\* Uplink workflow: A set of traces is “connected” to another set. This typically happens, when “external contributions” are imported (e.g. a team that proposes the interfaces of a subsystem). The connection links are created as proposals. For all new connection links a trace change workflow is executed. Redundancies are analysed and eliminated, what again triggers trace change workflows.\n\n\* Trace analysis workflow: Progress and trace consistency are analysed and reported by the modelling service\n\nThese workflows are assigned by the team or person, where the triggering work item is assigned to. The trace analysis is done by the modelling service on a frequent basis."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:GRANULARITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The granularity characterises the level of modularity of a system. The more subsystems a system is composed of, the higher its granularity.", -- "The granularity characterizes the level of modularity of a system. The more granularity is achieved, the more the system is decomposed in many sub-systems."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:GROUND--FOOTPRINT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The ground footprint of an object moving on the earth is defined as the projection of its volume on the earth ground.
\nNote: This defintion is applicable for collission detection between two ground vehicles moving on the same level.", -- "The ground footprint of an object moving on the earth is defined as the projection of its volume on the earth ground.\n\nNote: This defintion is applicable for collission detection between two ground vehicles moving on the same level."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HAZARD--MITIGATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Safety requirement regarding activities for lowering the effect of a hazard. It reduces the damage to people, property, and the environment by lowering the frequency, damping the consequences.
\nNote: A mitigation can be a realized by
\n
\n\* fault detection (hazard will be detected), typically this needs additional activities
\n\* failure reaction (after detection the hazard will be excluded), typically this needs additional activities
\n\* SRAC safety application condition (hazard will be avoided principly)", -- "Safety requirement regarding activities for lowering the effect of a hazard. It reduces the damage to people, property, and the environment by lowering the frequency, damping the consequences."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HAZARD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ Condition that could lead to an accident
\n Note 1 to entry: The equivalent definition in [IEC 60050-903:2013, 903-01-02] refers to "harm" instead of "accident".
\n Note 2 to entry: A Hazard sits at the boundary of the system under consideration. [ ERA-REC-116-2015-GUI] """, -- """ Condition that could lead to an accident \n\n Note 1 to entry: The equivalent definition in [IEC 60050-903:2013, 903-01-02] refers to "harm" instead of "accident".\n\n Note 2 to entry: A Hazard sits at the boundary of the system under consideration. [ ERA-REC-116-2015-GUI] """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HAZARD\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ Condition that could lead to an accident
\nNote 1 to entry: The equivalent definition in [IEC 60050-903:2013, 903-01-02] refers to "harm" instead of "accident".
\nNote 2 to entry: A Hazard sits at the boundary of the system under consideration. [ ERA-REC-116-2015-GUI] """, -- """ Condition that could lead to an accident \n\nNote 1 to entry: The equivalent definition in [IEC 60050-903:2013, 903-01-02] refers to "harm" instead of "accident".\n\nNote 2 to entry: A Hazard sits at the boundary of the system under consideration. [ ERA-REC-116-2015-GUI] """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HAZOP\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A hazard and operability study (HAZOP) is a structured and systematic examination of a complex planned or existing process or operation in order to identify and evaluate problems that may represent risks to personnel or equipment. The intention of performing a HAZOP is to review the design to pick up design and engineering issues that may otherwise not have been found. The technique is based on breaking the overall complex design of the process into a number of simpler sections called 'nodes' which are then individually reviewed. It is carried out by a suitably experienced multi-disciplinary team (HAZOP) during a series of meetings. The HAZOP technique is qualitative, and aims to stimulate the imagination of participants to identify potential hazards and operability problems. Structure and direction are given to the review process by applying standardised guide-word prompts to the review of each node.
\n[SOURCE: Wikipedia Hazard and operability study - Wikipedia]", -- "A hazard and operability study (HAZOP) is a structured and systematic examination of a complex planned or existing process or operation in order to identify and evaluate problems that may represent risks to personnel or equipment. The intention of performing a HAZOP is to review the design to pick up design and engineering issues that may otherwise not have been found. The technique is based on breaking the overall complex design of the process into a number of simpler sections called 'nodes' which are then individually reviewed. It is carried out by a suitably experienced multi-disciplinary team (HAZOP) during a series of meetings. The HAZOP technique is qualitative, and aims to stimulate the imagination of participants to identify potential hazards and operability problems. Structure and direction are given to the review process by applying standardised guide-word prompts to the review of each node.\n\n[SOURCE: Wikipedia Hazard and operability study - Wikipedia]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HFI--ACTIVITIES\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "HFI Activities are conducted to optimise the effectiveness and efficiency of human performance by fully considering the human contribution to system performance.
\n Human Performance is the observable and measurable behaviour that occurs in job and task situations, the extent to which goals such as speed, accuracy, quality and other criteria are met by people in their work environments.
\n The focus is on the ability of operators and maintainers to meet system performance requirements, including reliability and maintainability, under the conditions in which the system is employed.", -- "HFI Activities are conducted to optimise the effectiveness and efficiency of human performance by fully considering the human contribution to system performance.\n\n Human Performance is the observable and measurable behaviour that occurs in job and task situations, the extent to which goals such as speed, accuracy, quality and other criteria are met by people in their work environments.\n\n The focus is on the ability of operators and maintainers to meet system performance requirements, including reliability and maintainability, under the conditions in which the system is employed."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HSI\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Human-System Integration, an interdisciplinary approach that aims at optimizing in an early development stage the global system performance by desgining solutions adapted for both human and technical system.
\nHSI studies integrates work from multiple human centered domains - process analysis, both qualitative and quantitatve, ergonomics, safety, survivability, habitability, skill analysis, training.
\nHSI typically help define user interfaces and lead to workitem SPPR-2246 - Application Condition or SPPR-2244 - SRAC .", -- "Human-System Integration, an interdisciplinary approach that aims at optimizing in an early development stage the global system performance by desgining solutions adapted for both human and technical system. \n\nHSI studies integrates work from multiple human centered domains - process analysis, both qualitative and quantitatve, ergonomics, safety, survivability, habitability, skill analysis, training. \n\nHSI typically help define user interfaces and lead to workitem SPPR-2246 - Application Condition or SPPR-2244 - SRAC ."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HUMAN--FACTORS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Human factors issues include anything that affects human performance, particularly those factors that may cause or contribute to human error. The main human factors areas include:
\n
\n\* Individual (physical size, personal preferences, cognitive skills, attitudes, background)
\n\* Organisation (culture, work pattern, communications, supervision, training)
\n\* Job (design of equipment, rules and procedures, tools, signage, environment)
\nThe purpose of human factors is to minimise safety risk from the possibility of human error by:
\n
\n\* ensuring human characteristics are accounted for in the design (or re-design) of new and existing systems and equipment
\n\* identifying the issues which may cause or contribute to human errors
\n\* conducting activities and applying controls to reduce likelihood and consequences.", -- "Human factors issues include anything that affects human performance, particularly those factors that may cause or contribute to human error. The main human factors areas include:\n\n \n\n\* Individual (physical size, personal preferences, cognitive skills, attitudes, background)\n\n\* Organisation (culture, work pattern, communications, supervision, training)\n\n\* Job (design of equipment, rules and procedures, tools, signage, environment)\n\nThe purpose of human factors is to minimise safety risk from the possibility of human error by:\n\n \n\n\* ensuring human characteristics are accounted for in the design (or re-design) of new and existing systems and equipment\n\n\* identifying the issues which may cause or contribute to human errors\n\nconducting activities and applying controls to reduce likelihood and consequences."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:HUMAN--MACHINE--INTERFACE\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The Human Machine Interface is a physical interface permitting an interaction between a human and the system control. The purpose of this interface is to allow effective operation by the human (ie: acknowledgement for a function) permitting to the machine (computer) to adapt simultanously the behaviour of the control command.", -- "Graphical component to interact with a user e.g., by train graph."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:IMPERSONATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Impersonation, in terms of behavior, involves a person or entity imitating or mimicking the actions, mannerisms, or characteristics of another individual or entity.
\nIn the context on how this definition is to be understood, a newer system would behave exactly like an older version system of itself, allowing it to interact with older systems/subsystems transparently, by behaving exactly like the older version expected by them.
\n
\nFrom a software perspective, the different impersonations a system can take could be implemented as independent modules, which would reduce the complexity of each module implementing an specific impersonation, and also limit the side effects of changes or bugs in one module to the others.
\nIt could also facilitate the addition of a new version of a system, while still keeping the former version of the system available in a transparent way.", -- "Impersonation, in terms of behavior, involves a person or entity imitating or mimicking the actions, mannerisms, or characteristics of another individual or entity.\n\nIn the context on how this definition is to be understood, a newer system would behave exactly like an older version system of itself, allowing it to interact with older systems/subsystems transparently, by behaving exactly like the older version expected by them.\n\n\n\nFrom a software perspective, the different impersonations a system can take could be implemented as independent modules, which would reduce the complexity of each module implementing an specific impersonation, and also limit the side effects of changes or bugs in one module to the others.\n\nIt could also facilitate the addition of a new version of a system, while still keeping the former version of the system available in a transparent way."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:INPUT--DOCUMENTS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Input documents” are for example
\n \* Imported documents that are stored as “unstructured” content in the ALM in the first step (no work items created already)
\n \* Unstructured documents that are step by step converted to documents with structured work items
\n \* A model or model sketch from another modelling system, that is not compliant or convertible to SP modelling method/system, created as input document with concrete Polarion work items in the status “proposal”. These are seen as input sketches and are added manually into the SP modelling system using the correct method and modelling process
\n \* ALM documents in which new work items are created in a free structure and sequence (workflow designed in the domains)
\n\* Input documents are not used as formal deliverable (because they are created manually and may not contain all work items in a certain scope), but they can be used as background documentation.
\n\* The differentiation of input and output documents allows to create input documents along the needs of the workflows – having “things at one place where the team is currently working” – without the constraint to structure or “sort” it as a formal output
\n(image: 1-screenshot-20221230-141047.png)", -- "\* Input documents” are for example\n\n \* Imported documents that are stored as “unstructured” content in the ALM in the first step (no work items created already)\n\n \* Unstructured documents that are step by step converted to documents with structured work items\n\n \* A model or model sketch from another modelling system, that is not compliant or convertible to SP modelling method/system, created as input document with concrete Polarion work items in the status “proposal”. These are seen as input sketches and are added manually into the SP modelling system using the correct method and modelling process\n\n \* ALM documents in which new work items are created in a free structure and sequence (workflow designed in the domains)\n\n\* Input documents are not used as formal deliverable (because they are created manually and may not contain all work items in a certain scope), but they can be used as background documentation.\n\n\* The differentiation of input and output documents allows to create input documents along the needs of the workflows – having “things at one place where the team is currently working” – without the constraint to structure or “sort” it as a formal output\n\n(image: 1-screenshot-20221230-141047.png)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:INTEGRATION--TASK\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Combine elements, validate/test whole system behaviour, implement/commission
\n\* Right side of CENELEC “V”", -- "\* Combine elements, validate/test whole system behaviour, implement/commission\n\n\* Right side of CENELEC “V”"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:INTERCHANGEABILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Interchangeability is the ability to replace a subsystem from supplier A by a sub-system from supplier B without affecting other sub-systems or the overall system/subsystem and without any integration effort (lowest reasonable integration effort) and without any need for recertification. Exchangeability and interchangeability are related to the physical characteristics of sub-systems whereas interoperability is related to interactions between subsystems (e.g. also between STM and ETCS on-board there is interoperability).", -- "Interchangeability is the ability to replace a subsystem from supplier A by a sub-system from supplier B without affecting other sub-systems or the overall system/subsystem and without any integration effort (lowest reasonable integration effort) and without any need for recertification. Exchangeability and interchangeability are related to the physical characteristics of sub-systems wheres interoperability is related to interactions between subsystems (e.g. also between STM and ETCS on-board there is interoperability)."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:INTERFACE\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Based on ISO/IEC 2382, a shared boundary between two systems or subsystems, defined by various characteristics pertaining to the functions, physical signal exchanges, and other characteristics.", -- "With an interface the sub-systems of different suppliers are combined."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:INTERFACE\_lexConcept\_3 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "a means provided or required by components for an interaction between them or with actors", -- "Based on ISO/IEC 2382, a shared boundary between two systems or subsystems, defined by various characteristics pertaining to the functions, physical signal exchanges, and other characteristics."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:INTEROPERABILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Interoperability means the ability to allow the safe and uninterrupted movement of trains that accomplish the specified levels of performance.Reference to the TSI (European Regulation) would be good to understand the importance (double. needs to be deleted)", -- "Interoperability means the ability to allow the safe and uninterrupted movement of trains that accomplish the specified levels of performance. [Subset-023], [IOP-Dir 2016/797] so that a train is able to run across different infrastructure networks (IMs) and that an infrastructure network is able to interact with trains of different Railway Undertakings, using systems/sub-systems from different origins. Exchangeability and interchangeability are related to the physical characteristics of sub-systems wheres interoperability is related to interactions between subsystems (e.g. also between STM and ETCS on-board there is interoperability)."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:INTEROPERABILITY\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Interoperability means the ability to allow the safe and uninterrupted movement of trains that accomplish the specified levels of performance, see [Subset-023] SPT2ARC-1619 and [IOP-Dir 2016/797] SPT2ARC-1617 so that a train is able to run across different infrastructure networks (IMs) and that an infrastructure network is able to interact with trains of different Railway Undertakings, using systems/sub-systems from different origins. Exchangeability and interchangeability are related to the physical characteristics of sub-systems whereas interoperability is related to interactions between subsystems (e.g. also between STM and ETCS on-board there is interoperability).", -- "Interoperability means the ability to allow the safe and uninterrupted movement of trains that accomplish the specified levels of performance.Reference to the TSI (European Regulation) would be good to understand the importance (double. needs to be deleted)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:LEXICAL--GLOSSARY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "either imported from outside the project or authored inside the project, lexical glossaries are the containers for lexical entiries and abbreviations. A lexical glossary is first consolidated by its glossary author in its own scope, i.e. its definitions shall not be redundant with each other. It is then consolidated towards other glossaries. This glossary is then called SPPR-5441 - Missing cross-reference.
\nLexical glossaries are the source of the lexical definitons and abbreviations referred to in SPPR-5434 - Refering document.", -- "either imported from outside the project or authored inside the project, lexical glossaries are the containers for lexical entiries and abbreviations. A lexical glossary is first consolidated by its glossary author in its own scope, i.e. its definitions shall not be redundant with each other. It is then consolidated towards other glossaries. This glossary is then called SPPR-5441 - Missing cross-reference.\n\nLexical glossaries are the source of the lexical definitons and abbreviations referred to in SPPR-5434 - Refering document."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:LOGIAL--COMPONENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "conceptual element being part of the system, contributing to its behaviour, by interacting with other logical
\n components and logical actors", -- "conceptual element being part of the system, contributing to its behaviour, by interacting with other logical \n\n components and logical actors"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:LOGISTIC--DELAY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "delay, excluding administrative delay, incurred for the provision of resources needed for a maintenance action to proceed or continue.
\n[SOURCE: IEC 60050-192:2015, 192-07-13]", -- "delay, excluding administrative delay, incurred for the provision of resources needed for a maintenance action to proceed or continue.\n\n[SOURCE: IEC 60050-192:2015, 192-07-13]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MACMT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean active corrective maintenance time
\n expectation of the active corrective maintenance time
\n[SOURCE: IEC 60050-192:2015, 192-07-22]", -- "mean active corrective maintenance time\n\n expectation of the active corrective maintenance time\n\n[SOURCE: IEC 60050-192:2015, 192-07-22]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MAD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean administrative delay. Expectation of the administrative delay.
\n[SOURCE: IEC 60050-192:2015,192-07-26]", -- "mean administrative delay. Expectation of the administrative delay.\n\n[SOURCE: IEC 60050-192:2015,192-07-26]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MAINTAINABILITY--%3COF--AN--ITEM%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Ability to be retained in, or restored to, a state to perform as required, under given conditions of use and maintenance
\nNote 1 to entry: Given conditions would include aspects that affect maintainability, such as: location for maintenance, accessibility, maintenance procedures and maintenance resources.
\n[SOURCE: IEC 60050-192:2015, 192-01-27]
\nSource: SPPRAMSS-349 - [EN 50126-1:2017]", -- "Ability to be retained in, or restored to, a state to perform as required, under given conditions of use and maintenance \n\nNote 1 to entry: Given conditions would include aspects that affect maintainability, such as: location for maintenance, accessibility, maintenance procedures and maintenance resources. \n\n[SOURCE: IEC 60050-192:2015, 192-01-27] \n\nSource: SPPRAMSS-349 - [EN 50126-1:2017]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MAINTAINABILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Ability to be retained in, or restored to, a state to perform as required, under given conditions of use and maintenance [EN50126] SPT2ARC-1616 .", -- "Ability to be retained in, or restored to, a state to perform as required, under given conditions of use and maintenance [EN50126]."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MANAGEMENT--BY--TRACEABILITY--KANBAN--BASED\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A trace (nodes/work items and their links) is a graph that works out one issue (e.g. one requirement) in the most efficient, complete and fast way, down to its operational and system implementation
\n
\n (image: 1-image10.png)", -- "A trace (nodes/work items and their links) is a graph that works out one issue (e.g. one requirement) in the most efficient, complete and fast way, down to its operational and system implementation\n\n \n\n (image: 1-image10.png)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MDBF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Mean Distance Between Failures
\n [SOURCE: Wikipedia Mean Distance Between Failure – Wikipedia]", -- "Mean Distance Between Failures\n\n [SOURCE: Wikipedia Mean Distance Between Failure – Wikipedia]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MDBSF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Mean Distance Between Service Failures
\n [SOURCE: Wikipedia Mean Distance Between Failure – Wikipedia]", -- "Mean Distance Between Service Failures\n\n [SOURCE: Wikipedia Mean Distance Between Failure – Wikipedia]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:METHODOLOGY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ The ARCADIA method is used to identify functional chains, their overlapping scenarios and desired performance, along with their support by the architecture.
\n
\n\* Starting with the first level of system analysis, they ensure traceability throughout the process definition and check each proposed architectural design against expected performance and constraints.
\n\* The non-functional properties expected from the system solution are also formalized in 'viewpoints'. Each viewpoint captures constraints that the system should face or meet (feared events, security threats, latency expectations, product line or reuse constraints, power consumption or cost issues, and more).
\n\* Then the architecture model is automatically analyzed to verify that it meets these constraints, thanks to dedicated expert rules (performance computation, resource consumption, safety or security barriers, etc.). This analysis can be done very early in the development cycle, detecting design issues as soon as possible ("early validation").
\nAs a summary, the approach to characterization by views (or "viewpoints") cross-checks that the proposed architecture is capable of providing the required functions with the desired level of performance, security, dependability, mass, scalability, environments, mass, interfaces, etc. ensuring the consistency of engineering decisions, because all engineering stakeholders share the same engineering information, and can apply his/her own views and checks to them, so as to secure the common definition. """, -- """ The ARCADIA method is used to identify functional chains, their overlapping scenarios and desired performance, along with their support by the architecture. \n\n \n\n\* Starting with the first level of system analysis, they ensure traceability throughout the process definition and check each proposed architectural design against expected performance and constraints.\n\n\* The non-functional properties expected from the system solution are also formalized in 'viewpoints'. Each viewpoint captures constraints that the system should face or meet (feared events, security threats, latency expectations, product line or reuse constraints, power consumption or cost issues, and more).\n\n\* Then the architecture model is automatically analyzed to verify that it meets these constraints, thanks to dedicated expert rules (performance computation, resource consumption, safety or security barriers, etc.). This analysis can be done very early in the development cycle, detecting design issues as soon as possible ("early validation").\n\nAs a summary, the approach to characterization by views (or "viewpoints") cross-checks that the proposed architecture is capable of providing the required functions with the desired level of performance, security, dependability, mass, scalability, environments, mass, interfaces, etc. ensuring the consistency of engineering decisions, because all engineering stakeholders share the same engineering information, and can apply his/her own views and checks to them, so as to secure the common definition. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MFDT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean fault detection time
\n[SOURCE: SPPRAMSS-3539 - [EN 61703: 2016]]", -- "mean fault detection time \n\n[SOURCE: SPPRAMSS-3539 - [EN 61703: 2016]]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MLD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean logistic delay. Expectation of the logistic delay.
\n[SOURCE: IEC 60050-192:2015,192-07-27]", -- "mean logistic delay. Expectation of the logistic delay.\n\n[SOURCE: IEC 60050-192:2015,192-07-27]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MODE--TRANSITION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A change of mode toward itself or toward another mode. A transition is characterised by a trigger which can be a boolean
\n condition applied on certain model elements, such as functional exchanges.", -- "A change of mode toward itself or toward another mode. A transition is characterised by a trigger which can be a boolean \n\n condition applied on certain model elements, such as functional exchanges."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MODEL--SYNCHRONISATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* There are two types of synchronisations:
\n \* A copy of all MBSE models is stored in the ALM. The copied work items can be part of the ALM workflow or can be approved, decided, commented and linked in the ALM.
\n \* Synchronisation between “mother LA/PA” and “daughter LA/PA” in the MBSE platform. These are done to split architecting work to different teams.
\n \* Example (start situation for the SP):
\n(image: 1-Picture\_1.png)", -- "\* There are two types of synchronisations:\n\n \* A copy of all MBSE models is stored in the ALM. The copied work items can be part of the ALM workflow or can be approved, decided, commented and linked in the ALM.\n\n \* Synchronisation between “mother LA/PA” and “daughter LA/PA” in the MBSE platform. These are done to split architecting work to different teams.\n\n \* Example (start situation for the SP):\n\n(image: 1-Picture\_1.png)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MODULARITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Modularity is used in this document as a general term for dividing a system/sub-system/module in sub-systems/modules.", -- "Modularity is used in this document as a general term for devide a system/sub-system/module in sub-systems/modules."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MODULARITY\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The property of a system being composed of a coherent whole of single, independent building blocks or modules.", -- "Modularity is used in this document as a general term for dividing a system/sub-system/module in sub-systems/modules."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MODULE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "{comment:34}
\nA module is a sourceable unit of the CCS system (hardware and/or software), having full defined functionality, interface, performance, safety. {comment:180}", -- "{comment:34}\n\nA module is a sourceable unit of the CCS system (hardware and/or software), having full defined functionality, interface, performance, safety. {comment:180}"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MOE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Measure of Effectiveness (that the system performs as required). Characteristics:
\n\* Relates to performance
\n\* Objective
\n\* Simple to state
\n\* Testable
\n\* Complete
\n\* Clear
\n\* States any time dependency
\n\* States any environmental conditions
\n\* Can be measured quantitatively (if required, may be measured statistically or as a probability)
\n\* Easy to measure
\n\* Select only MoEs that measure the degree to which the desired outcome is achieved
\n\* Use the same MoEs to measure more than one condition when appropriate
\n\* Structure so that they have measurable, collectible, and relevant indicators
\n\* Write as statements (not questions)
\n\* Maximize clarity", -- "Measure of Effectiveness (that the system performs as required). Characteristics: \n\n\* Relates to performance\n\n\* Objective\n\n\* Simple to state\n\n\* Testable\n\n\* Complete\n\n\* Clear\n\n\* States any time dependency\n\n\* States any environmental conditions\n\n\* Can be measured quantitatively (if required, may be measured statistically or as a probability)\n\n\* Easy to measure\n\n\* Select only MoEs that measure the degree to which the desired outcome is achieved\n\n\* Use the same MoEs to measure more than one condition when appropriate\n\n\* Structure so that they have measurable, collectible, and relevant indicators\n\n\* Write as statements (not questions)\n\n\* Maximize clarity"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MOTBF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean operating time between failures
\n expectation of the duration of the operating time between failures
\n Note 1 to entry: Mean operating time between failures should only be applied to repairable items. For non-repairable items, see mean operating time to failure (192-05-11) SPPRAMSS-4040 - MTTF .
\n[SOURCE: IEC 60050-192:2015, 192-05-13]", -- "mean operating time between failures\n\n expectation of the duration of the operating time between failures\n\n Note 1 to entry: Mean operating time between failures should only be applied to repairable items. For non-repairable items, see mean operating time to failure (192-05-11) SPPRAMSS-4040 - MTTF .\n\n[SOURCE: IEC 60050-192:2015, 192-05-13]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MRT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "expectation of the (mean) repair time
\n[SOURCE: IEC 60050-192:2015, 192-07-21]
\nNote 1: MRT = fault localization time + fault correction time + function checkout time according SPPRAMSS-3539 - [EN 61703: 2016]", -- "expectation of the (mean) repair time\n\n[SOURCE: IEC 60050-192:2015, 192-07-21]\n\nNote 1: MRT = FLT + FCT + FCKT according SPPRAMSS-3539 - [EN 61703: 2016]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MTBF\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean operating time between failures
\n expectation of the duration of the operating time between failures
\n Note 1 to entry: Mean operating time between failures should only be applied to repairable items. For non-repairable items, see mean operating time to failure (192-05-11) SPPRAMSS-4040 - MTTF .
\n[SOURCE: IEC 60050-192:2015, 192-05-13]", -- "mean operating time between failures\n\n expectation of the duration of the operating time between failures\n\n Note 1 to entry: Mean operating time between failures should only be applied to repairable items. For non-repairable items, see mean operating time to failure (192-05-11) SPPRAMSS-4040 - MTTF .\n\n[SOURCE: IEC 60050-192:2015, 192-05-13]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MTBSF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean operating time between service failures
\n Note 1 to entry: Service failures should be defined as one of the failures severity levels. These failures are understood as failures with operational impact. For example failures for which the safety reaction is application of the service brake.
\n Note 2 to entry: Abbreviation is not yet defined in any CEN/CENELEC railway standard but is often used by railway operators and infrastructure managers.", -- "Mean (operating) Time Between Service Failures\n\n Note: it is often used by customers to distinguish the failure classes with operational impact from intrinsic ones. It is not coming from a standard."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MTD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean technical delay
\n[SOURCE: SPPRAMSS-349 - [EN 50126-1:2017] Annex B.4]", -- "mean technical delay \n\n[SOURCE: SPPRAMSS-349 - [EN 50126-1:2017] Annex B.4]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MTTFF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean operating time to first failure
\nexpectation of the operating time to first failure
\nNote 1 to entry: See also operating time to first failure (192-05-02).
\nNote 2 to entry: For non-repairable items, the MTTFF is also the MTTF.
\n[SOURCE: IEC 60050-192:2015, 192-05-12]", -- "mean operating time to first failure\n\nexpectation of the operating time to first failure\n\nNote 1 to entry: See also operating time to first failure (192-05-02).\n\nNote 2 to entry: For non-repairable items, the MTTFF is also the MTTF.\n\n[SOURCE: IEC 60050-192:2015, 192-05-12]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MTTF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ mean operating time to failure
\n expectation of the operating time to failure
\n Note 1 to entry: In the case of non-repairable items with an exponential distribution of operating times to failure (i.e. a constant failure rate) the MTTF is numerically equal to the reciprocal of the failure rate. This is also true for repairable items if after restoration they can be considered to be "as-good-as-new".
\n Note 2 to entry: See also operating time to failure (192-05-01) SPPRAMSS-4441 - operating time to failure, <of an item> .
\n[SOURCE: IEC 60050-192:2015, 192-05-11] """, -- """ mean operating time to failure\n\n expectation of the operating time to failure\n\n Note 1 to entry: In the case of non-repairable items with an exponential distribution of operating times to failure (i.e. a constant failure rate) the MTTF is numerically equal to the reciprocal of the failure rate. This is also true for repairable items if after restoration they can be considered to be "as-good-as-new".\n\n Note 2 to entry: See also operating time to failure (192-05-01) SPPRAMSS-4441 - operating time to failure, <of an item> .\n\n[SOURCE: IEC 60050-192:2015, 192-05-11] """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:MTTR\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "mean time to restoration - expectation of the time to restoration
\n deprecated: mean time to repair, mean time to recovery
\n Note 1 to entry: IEC 60050-191:1990 (now withdrawn; replaced by IEC 60050-192:2015) defined the term ”mean time to recovery” as a synonym, but restoration and recovery are not synonyms.
\n Note 2 to entry: MTTR = MFDT + MAD + MLD + MTD + MRT according SPPRAMSS-3539 - [EN 61703: 2016]
\n [SOURCE: IEC 60050-192, 192-07-23, modified: Note 2 to entry added.]", -- "Mean time to restoration\n\n [SOURCE: SPPRAMSS-349 - [EN 50126-1:2017] Annex B.4]\n\n Note 1: MTTR = MFDT + MAD + MLD + MTD + MRT according SPPRAMSS-3539 - [EN 61703: 2016]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:NATIONAL--IMPLEMENATION--PLAN\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "In 2016, the EC also asked the EU Member States to establish the so-called “National Implementation Plans” (NIPs) in which they have to describe their actions to comply with the relevant standard for ERTMS, i.e. CCS TSI 2016/919. The CCS TSI regulates the implementation of fully interoperable 'control-command and signalling' subsystems. The NIPs have to fulfil two conditions, i.e. they must cover a period of at least 15 years and they must be updated every 5 years. In addition, they have to contain the following information:
\n
\n (image: 1-screenshot-20230616-124455.png)", -- "In 2016, the EC also asked the EU Member States to establish the so-called “National Implementation Plans” (NIPs) in which they have to describe their actions to comply with the relevant standard for ERTMS, i.e. CCS TSI 2016/919. The CCS TSI regulates the implementation of fully interoperable 'control-command and signalling' subsystems. The NIPs have to fulfil two conditions, i.e. they must cover a period of at least 15 years and they must be updated every 5 years. In addition, they have to contain the following information: \n\n\n\n (image: 1-screenshot-20230616-124455.png)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:NON-FUNCTIONAL--REQUIREMENTS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Non-Functional Requirements are constraints on the system that define how well something is to be done or how it is to be done which fall into three categories:{comment:66}
\n\* Performance Requirements SPPR-4175 - Performance Requirements ;
\n\* System Requirements SPPR-4177 - System Requirements ;{comment:65}
\n\* Implementation Requirements SPPR-4180 - Implementation Requirements .
\nThese categories are derived from consideration of the essential relationships with the Operational{comment:31} and Functional Requirements and ensure the completeness and consistency of the requirements model (It is these essential relationships between the functionally-based requirements and constraint-based requirements that make the Holistic Requirements Model so useful).", -- "Non-Functional Requirements are constraints on the system that define how well something is to be done or how it is to be done which fall into three categories:{comment:66} \n\n\* Performance Requirements SPPR-4175 - Performance Requirements ;\n\n\* System Requirements SPPR-4177 - System Requirements ;{comment:65}\n\n\* Implementation Requirements SPPR-4180 - Implementation Requirements .\n\nThese categories are derived from consideration of the essential relationships with the Operational{comment:31} and Functional Requirements and ensure the completeness and consistency of the requirements model (It is these essential relationships between the functionally-based requirements and constraint-based requirements that make the Holistic Requirements Model so useful)."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:NOTIF-IT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The database for Notified National Rules. Has been replaced by SRD: Single Rule Database
\n[SOURCE: ERA, https://www.era.europa.eu/domains/registers/srd\_en ]", -- "The database for Notified National Rules. Has been replaced by SRD: Single Rule Database\n\n[SOURCE: ERA, https://www.era.europa.eu/domains/registers/srd\_en ]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OAB\_--\_ENTITIES\_ACTORS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capella Viewpoint.
\nPurpose: displays the sub-entities and actors of an entity.
\nRule: contains at least 1 operational entity
\nRule: May contain operational actors
\nRule: contains no operational activities
\nFilter: TBD", -- "Capella Viewpoint.\n\nPurpose: displays the sub-entities and actors of an entity.\n\nRule: contains at least 1 operational entity\n\nRule: May contain operational actors \n\nRule: contains no operational activities\n\nFilter: TBD"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OAB\_--\_PROCESS--INVOLVEMENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capella Viewpoint.
\nPurpose: For a process p, displays the activities it involves and the entities or actors they are allocated to.
\nRule0: There is one and only one [OAB][ProcessInvolvement] per process
\nRule1: At least each leaf activity/actor hosting the an activity of p shall be displayed on the diagram.
\nRule2: Structure Entities, defined as containing the activity or actors or rule 1, may be displayed (Author's decision). If so, the diagram shall remain clear even if exported on A4 protrait.
\nRule3: Contains at least 1 operational entity, at least 2 actitivies.
\nRule4: The diagram forbids entities or actors that do not allocate any activity of the process it describes
\nFilter: TBD", -- "Capella Viewpoint.\n\nPurpose: For a process p, displays the activities it involves and the entities or actors they are allocated to. \n\nRule0: There is one and only one [OAB][ProcessInvolvement] per process\n\nRule1: At least each leaf activity/actor hosting the an activity of p shall be displayed on the diagram.\n\nRule2: Structure Entities, defined as containing the activity or actors or rule 1, may be displayed (Author's decision). If so, the diagram shall remain clear even if exported on A4 protrait.\n\nRule3: Contains at least 1 operational entity, at least 2 actitivies.\n\nRule4: The diagram forbids entities or actors that do not allocate any activity of the process it describes\n\nFilter: TBD"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OAIB\_--\_OPERATIONAL--PROCESS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capella Viewpoint.
\nPurpose: displays the activities of a process.
\nRule: contains at least 2 operational activity
\nRule: each activity has an interaction with another activity
\nFilter: TBD", -- "Capella Viewpoint.\n\nPurpose: displays the activities of a process.\n\nRule: contains at least 2 operational activity\n\nRule: each activity has an interaction with another activity \n\nFilter: TBD"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OBJECTIVES\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ ARCADIA is a structured engineering method to identify and check the architecture of complex systems. It promotes collaborative work among all stakeholders during many of the engineering phases of the system. It allows iterations during the definition phase that help the architects to converge towards satisfaction of all identified needs. Even if textual requirements are kept as a support for part of customer need capture, ARCADIA favors functional analysis as the major way to formalize the need and solution behavior. This includes operational, functional and non-functional aspects, along with resulting definition of the architecture, based on – and justified against – this functional analysis. ARCADIA is based on the following general principles
\n\* All engineering stakeholders share the same language, method set of engineering artifacts and information, description of the need and the product itself as a shared model;
\n\* Each set of constraints (e.g. safety, performance, cost, mass, etc.) is formalized in a "viewpoint" against which each candidate architecture will be checked;
\n\* Architecture verification rules are established and the model is challenged against them, so as to check that architecture definition meets expectations, as early as possible in the process;
\n\* Co-engineering between the different levels of engineering is supported by the joint development of models. Models of various levels of the architecture and trade-offs are deduced, validated and/or connected with each other.
\nThe ARCADIA method is tooled through Capella, a modeling tool that meets full-scale deployment constraints in an operational context. Capella is available free of charge from the engineering community under open source. """, -- """ ARCADIA is a structured engineering method to identify and check the architecture of complex systems. It promotes collaborative work among all stakeholders during many of the engineering phases of the system. It allows iterations during the definition phase that help the architects to converge towards satisfaction of all identified needs. Even if textual requirements are kept as a support for part of customer need capture, ARCADIA favors functional analysis as the major way to formalize the need and solution behavior. This includes operational, functional and non-functional aspects, along with resulting definition of the architecture, based on – and justified against – this functional analysis. ARCADIA is based on the following general principles \n\n\* All engineering stakeholders share the same language, method set of engineering artifacts and information, description of the need and the product itself as a shared model;\n\n\* Each set of constraints (e.g. safety, performance, cost, mass, etc.) is formalized in a "viewpoint" against which each candidate architecture will be checked;\n\n\* Architecture verification rules are established and the model is challenged against them, so as to check that architecture definition meets expectations, as early as possible in the process;\n\n\* Co-engineering between the different levels of engineering is supported by the joint development of models. Models of various levels of the architecture and trade-offs are deduced, validated and/or connected with each other.\n\nThe ARCADIA method is tooled through Capella, a modeling tool that meets full-scale deployment constraints in an operational context. Capella is available free of charge from the engineering community under open source. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OCB\_--\_INVOLVEMENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capella Viewpoint.
\nPurpose: displays the operational entities and actors involved in an operational capability- and only that
\nRule: contains 1 and only 1 capability
\nRule: contains at least on 1 entity or actor
\nFilter: show entities
\n
\n\* Yes: Hide Communication Means
\n\* No: Hide Involvement Links
\n\* Yes: Hide Operational Capabiltiy Extends
\n\* Yes: Hide Operational Capabiltiy Includes
\n\* Yes: Hide Operational Capabiltiy Generatlizations
\n\* Yes: Hide Property Values
\n\* Yes: Hide Diagram Title Blocks
\n\* Yes: Hilde Element Title Blocks", -- "Capella Viewpoint.\n\nPurpose: displays the operational entities and actors involved in an operational capability- and only that\n\nRule: contains 1 and only 1 capability\n\nRule: contains at least on 1 entity or actor\n\nFilter: show entities\n\n \n\n\* Yes: Hide Communication Means\n\n\* No: Hide Involvement Links\n\n\* Yes: Hide Operational Capabiltiy Extends\n\n\* Yes: Hide Operational Capabiltiy Includes\n\n\* Yes: Hide Operational Capabiltiy Generatlizations\n\n\* Yes: Hide Property Values\n\n\* Yes: Hide Diagram Title Blocks\n\n\* Yes: Hilde Element Title Blocks"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OCB\_--\_STRUCTURE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capella Viewpoint.
\nPurpose: displays the hierachy of operational capabilities - and only that
\nRule: contains only capabilities and their relationships, no entities
\nRule: organised as a tree (folder like)
\nFilter:
\n
\n\* Yes: Hide Communication Means
\n\* Yes: Hide Involvement Links
\n\* TBD: Hide Operational Capabiltiy Extends
\n\* TBD: Hide Operational Capabiltiy Includes
\n\* TBD: Hide Operational Capabiltiy Generatlizations
\n\* Yes: Hide Property Values
\n\* Yes: Hide Diagram Title Blocks
\n\* Yes: Hilde Element Title Blocks", -- "Capella Viewpoint.\n\nPurpose: displays the hierachy of operational capabilities - and only that\n\nRule: contains only capabilities and their relationships, no entities\n\nRule: organised as a tree (folder like)\n\nFilter:\n\n \n\n\* Yes: Hide Communication Means\n\n\* Yes: Hide Involvement Links\n\n\* TBD: Hide Operational Capabiltiy Extends\n\n\* TBD: Hide Operational Capabiltiy Includes\n\n\* TBD: Hide Operational Capabiltiy Generatlizations\n\n\* Yes: Hide Property Values\n\n\* Yes: Hide Diagram Title Blocks\n\n\* Yes: Hilde Element Title Blocks"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OES\_--\_OPERATIONAL--PROCESS\_--\_SCENARIO\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capella Viewpoint.
\nPurpose: Display the selected sequence of activities, interactions and related dependencies in a swimlane diagram
\nRule: Each activity is deployed to an entity or an actor.
\nRule: Each activity has at least one input and one output interaction.
\nFilter: TBD selected path/sequence", -- "Capella Viewpoint.\n\nPurpose: Display the selected sequence of activities, interactions and related dependencies in a swimlane diagram\n\nRule: Each activity is deployed to an entity or an actor.\n\nRule: Each activity has at least one input and one output interaction.\n\nFilter: TBD selected path/sequence"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OPD\_--\_OPERATIONAL--PROCESS\_--\_LOGIC\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Capella viewpoint
\nPurpose: Show all activities of operational process with their data flow (interactions) and control flow.
\nRule: Each activity has at least one input and one output interaction.
\nFilter: TBD", -- "Capella viewpoint\n\nPurpose: Show all activities of operational process with their data flow (interactions) and control flow.\n\nRule: Each activity has at least one input and one output interaction. \n\nFilter: TBD"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OPERATING--TIME--TO--FAILURE--%3COF--AN--ITEM%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "operating time accumulated from the first use, or from restoration, until failure
\n Note 1 to entry: See also operating time (192-02-05) SPPRAMSS-4080 - operating time .
\n [SOURCE: IEC 60050-192:2015, 192-05-01]", -- "operating time accumulated from the first use, or from restoration, until failure\n\n Note 1 to entry: See also operating time (192-02-05) SPPRAMSS-4080 - operating time .\n\n [SOURCE: IEC 60050-192:2015, 192-05-01]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OPERATING--TIME\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Time interval for which an item is in an operating state
\n Note 1 to entry: The duration of operating time may be expressed in units appropriate to the item concerned, e.g. calendar time, operating cycles, distance run, etc., and the units should always be clearly stated.
\n [SOURCE: IEC 60050-192:2015, 192-02-05]", -- "Time interval for which an item is in an operating state \n\n Note 1 to entry: The duration of operating time may be expressed in units appropriate to the item concerned, e.g. calendar time, operating cycles, distance run, etc., and the units should always be clearly stated.\n\n [SOURCE: IEC 60050-192:2015, 192-02-05]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:OPERATIONAL--HARMONIZATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ An operational topic (process, actor, defintion, ...) is considered harmonized if a description is provided in the SP context that fulfills:
\n
\n\* This description is exhaustive, delimited and precise. There is no remaining "open space and freedom". The scope is clearly defined. Existing gaps are clearly identified. Statements are formulated unambiguously, if necessary in a formal form. {comment:98}
\n\* A set of variants and parameters, if necessary, are part of the description.
\n\* The description has been agreed by both Railway and Industry stakeholdes.
\nThis consens is especially defined under the light of project rework and domain standardisaton:
\n
\n\* The description is agreed by the railways stakeholders of system Pillar as fitting their need, i.e. all their needs are fullfilled by the available variants and parameters without having again a solution discussion or check.
\n
\n\* This description is agreed by the industry stakeholdes as 1:1 implementable, without having to reopen again the ambiguity discussion """, -- """ An operational topic (process, actor, defintion, ...) is considered harmonized if a description is provided in the SP context that fulfills:\n\n \n\n\* This description is exhaustive, delimited and precise. There is no remaining "open space and freedom". The scope is clearly defined. Existing gaps are clearly identified. Statements are formulated unambiguously, if necessary in a formal form. {comment:98}\n\n\* A set of variants and parameters, if necessary, are part of the description.\n\n\* The description has been agreed by both Railway and Industry stakeholdes.\n\nThis consens is especially defined under the light of project rework and domain standardisaton:\n\n \n\n\* The description is agreed by the railways stakeholders of system Pillar as fitting their need, i.e. all their needs are fullfilled by the available variants and parameters without having again a solution discussion or check.\n\n \n\n\* This description is agreed by the industry stakeholdes as 1:1 implementable, without having to reopen again the ambiguity discussion """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:ORS--OPERATIONAL--REQUIREMENT--SPECIFICATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The Operational Requirement Specification consists of a document bundle published by System Pillar's OA (SPPR-4033 ) as depicted in Diagram SPPR-3655 - Sub-processes of the operational analysis and their work-products.
\n(TBD: detail while defining the documents created in Process SPPR-2240 - P2.5 Operational Design (System Level 2-3) )", -- "The Operational Requirement Specification consists of a document bundle published by System Pillar's OA (SPPR-4033 ) as depicted in Diagram SPPR-3655 - Sub-processes of the operational analysis and their work-products.\n\n(TBD: detail while defining the documents created in Process SPPR-2240 - P2.5 Operational Design (System Level 2-3) )"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:PERFORMANCE--%3C--OF--AN--ITEM--%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Performance is the ability of all technical and operational functions, in a specific operational context, to deliver a pre-defined and agreed level of service of the system and/or vehicle in operation.
\n
\n With the addition (and further clarification) that:
\n The minimum level of performance is determined and influenced by technical and operational aspects on system, vehicle and operational level, such as but not limited to: reliability, availability, maintainability, safety, security, human factors, quality, mission profile, environment, environmental conditions and laws and regulations. This means the scope and boundaries (span of control) of performance need to be defined clearly.
\n This also stresses why the inclusion of the principles of the RAM Policy to the ERJU activities for PRAMS, is of high importance.", -- "Performance is the ability of all technical and operational functions, in a specific operational context, to deliver a pre-defined and agreed level of service of the system and/or vehicle in operation.\n\n\n\n With the addition (and further clarification) that:\n\n The minimum level of performance is determined and influenced by technical and operational aspects on system, vehicle and operational level, such as but not limited to: reliability, availability, maintainability, safety, security, human factors, quality, mission profile, environment, environmental conditions and laws and regulations. This means the scope and boundaries (span of control) of performance need to be defined clearly.\n\n This also stresses why the inclusion of the principles of the RAM Policy to the ERJU activities for PRAMS, is of high importance."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:PFH\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ average frequency of a dangerous failure per hour
\n
\naverage frequency of a dangerous failure of an E/E/PE safety related system to perform the
\nspecified safety function over a given period of time
\n
\nNOTE 1 The term “probability of dangerous failure per hour” is not used in this standard but the acronym PFH has
\nbeen retained but when it is used it means “average frequency of dangerous failure [h]".
\nNOTE 2 From a theoretical point of view, the PFH is the average of the unconditional failure intensity, also called
\nfailure frequency, and which is generally designated w(t). It should not be confused with a failure rate (see Annex B
\nof IEC 61508-6).
\nNOTE 3 When the E/E/PE safety-related system is the ultimate safety layer, the PFH should be calculated from its
\nunreliability F(T)=1-R(t) (see “failure rate” above). When it is not the ultimate safety-related system its PFH should
\nbe calculated from its unavailability U(t) (see PFD above). PFH approximations are given by F(T)/T and 1/MTTF in
\nthe first case and 1/MTBF in the second case.
\nNOTE 4 When the E/E/PE safety-related system implies only quickly repaired revealed failures then an asymptotic
\nfailure rate λas is quickly reached. It provides an estimate of the PFH.
\n
\n[EN 61508-4:2010 - §3.6.19] """, -- """ average frequency of a dangerous failure per hour \n\n\n\naverage frequency of a dangerous failure of an E/E/PE safety related system to perform the \n\nspecified safety function over a given period of time \n\n\n\nNOTE 1 The term “probability of dangerous failure per hour” is not used in this standard but the acronym PFH has \n\nbeen retained but when it is used it means “average frequency of dangerous failure [h]". \n\nNOTE 2 From a theoretical point of view, the PFH is the average of the unconditional failure intensity, also called \n\nfailure frequency, and which is generally designated w(t). It should not be confused with a failure rate (see Annex B \n\nof IEC 61508-6). \n\nNOTE 3 When the E/E/PE safety-related system is the ultimate safety layer, the PFH should be calculated from its \n\nunreliability F(T)=1-R(t) (see “failure rate” above). When it is not the ultimate safety-related system its PFH should \n\nbe calculated from its unavailability U(t) (see PFD above). PFH approximations are given by F(T)/T and 1/MTTF in \n\nthe first case and 1/MTBF in the second case. \n\nNOTE 4 When the E/E/PE safety-related system implies only quickly repaired revealed failures then an asymptotic \n\nfailure rate λas is quickly reached. It provides an estimate of the PFH. \n\n\n\n[EN 61508-4:2010 - §3.6.19] """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:PFH\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "probability of dangerous failure per hour
\n[EN 61508-4:2010]", -- "probability of dangerous failure per hour\n\n[EN 61508-4:2010]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:PFH\_lexConcept\_3 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "probability of dangerous failure per hour
\n[SOURCE: EN 61508-4:2010]", -- "probability of dangerous failure per hour\n\n[SOURCE: EN 61508-4:2010]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:PHYSICAL--LINK\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "means of communication/transport relating to the hosting of physical components, used as a medium by exchanges
\n between behavioural components", -- "means of communication/transport relating to the hosting of physical components, used as a medium by exchanges \n\n between behavioural components"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:PHYSICAL--PORT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "connection point of one or more physical links allocated to a host component, by specifying how the component can be
\n connected to the others or external actors", -- "connection point of one or more physical links allocated to a host component, by specifying how the component can be \n\n connected to the others or external actors"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:PLATEAU\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ In the context of system engineering, a "plateau" refers to a distinct stage or phase within the system development lifecycle (SDLC) where progress or development remains relatively stable or flat for a period of time. It is characterized by a temporary pause or leveling off in the advancement or evolution of the system being developed.
\n
\n During a plateau, the system or project may have reached a certain level of functionality, maturity, or completion. It can be seen as a natural phase in the development process, where the initial rapid progress slows down, and further enhancements or major changes are temporarily put on hold. This can occur for various reasons, such as:
\n\* Technical Stability: The system has achieved a stable and functioning state, meeting the primary objectives or requirements defined for that particular stage of development. The focus during this phase may shift towards fine-tuning, optimization, and bug fixing.
\n\* Resource Allocation: The allocation of resources, including personnel, time, and budget, may be temporarily redirected to other areas of priority. This can result in a pause in major advancements for the system on the plateau.
\n\* Stakeholder Evaluation: Stakeholders may require time to evaluate and provide feedback on the current state of the system before proceeding to the next phase. This evaluation period can lead to a plateau in development activities.
\n\* Planning and Preparation: The system engineering team may use the plateau phase to plan and prepare for the next stage of development. This includes conducting feasibility studies, conducting risk assessments, gathering requirements, or developing a roadmap for future enhancements or releases.
\nWhile plateaus can be a temporary slowdown in the system development process, they can also provide opportunities for reflection, evaluation, and strategic decision-making. During this phase, system engineers and stakeholders can assess the current state of the system, address any outstanding issues or challenges, and plan for future iterations or developments.
\n
\n It's important for system engineers to effectively manage plateaus by maintaining clear communication with stakeholders, ensuring proper documentation, and utilizing the time and resources available to refine and improve the system before progressing to the next stage of development. """, -- """ In the context of system engineering, a "plateau" refers to a distinct stage or phase within the system development lifecycle (SDLC) where progress or development remains relatively stable or flat for a period of time. It is characterized by a temporary pause or leveling off in the advancement or evolution of the system being developed.\n\n\n\n During a plateau, the system or project may have reached a certain level of functionality, maturity, or completion. It can be seen as a natural phase in the development process, where the initial rapid progress slows down, and further enhancements or major changes are temporarily put on hold. This can occur for various reasons, such as: \n\n\* Technical Stability: The system has achieved a stable and functioning state, meeting the primary objectives or requirements defined for that particular stage of development. The focus during this phase may shift towards fine-tuning, optimization, and bug fixing.\n\n\* Resource Allocation: The allocation of resources, including personnel, time, and budget, may be temporarily redirected to other areas of priority. This can result in a pause in major advancements for the system on the plateau.\n\n\* Stakeholder Evaluation: Stakeholders may require time to evaluate and provide feedback on the current state of the system before proceeding to the next phase. This evaluation period can lead to a plateau in development activities.\n\n\* Planning and Preparation: The system engineering team may use the plateau phase to plan and prepare for the next stage of development. This includes conducting feasibility studies, conducting risk assessments, gathering requirements, or developing a roadmap for future enhancements or releases.\n\nWhile plateaus can be a temporary slowdown in the system development process, they can also provide opportunities for reflection, evaluation, and strategic decision-making. During this phase, system engineers and stakeholders can assess the current state of the system, address any outstanding issues or challenges, and plan for future iterations or developments.\n\n \n\n It's important for system engineers to effectively manage plateaus by maintaining clear communication with stakeholders, ensuring proper documentation, and utilizing the time and resources available to refine and improve the system before progressing to the next stage of development. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:RBD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "reliability block diagram
\nlogical, graphical representation of a system showing how reliability of its sub items (represented by blocks) and combinations thereof, affect reliability of the system
\n[SOURCE: IEC 60050-192:2015, 192-11-03]", -- "reliability block diagram\n\nlogical, graphical representation of a system showing how reliability of its sub items (represented by blocks) and combinations thereof, affect reliability of the system\n\n[SOURCE: IEC 60050-192:2015, 192-11-03]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:RELIABILITY--%3COF--AN--ITEM%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Ability to perform as required, without failure, for a given time interval, under given conditions.
\nNote 1 to entry: The time interval duration can be expressed in units appropriate to the item concerned, e.g.
\ncalendar time, operating cycles, distance run, etc.
\nNote 2 to entry: Given conditions include aspects that affect reliability, such as: mode of operation, stress levels,
\nenvironmental conditions, and maintenance.
\nNote 3 to entry: Reliability can be quantified using measures defined in Section 192-05, Reliability related
\nconcepts: measures.
\n[SOURCE: IEC 60050-192:2015, 192-01-24]
\nSource: SPPRAMSS-349 - [EN 50126-1:2017]", -- "Ability to perform as required, without failure, for a given time interval, under given conditions. \n\nNote 1 to entry: The time interval duration can be expressed in units appropriate to the item concerned, e.g. \n\ncalendar time, operating cycles, distance run, etc. \n\nNote 2 to entry: Given conditions include aspects that affect reliability, such as: mode of operation, stress levels, \n\nenvironmental conditions, and maintenance. \n\nNote 3 to entry: Reliability can be quantified using measures defined in Section 192-05, Reliability related \n\nconcepts: measures. \n\n[SOURCE: IEC 60050-192:2015, 192-01-24] \n\nSource: SPPRAMSS-349 - [EN 50126-1:2017]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:REQUIREMENT--STATEMENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A requirement statement is the result of a formal transformation of one or more needs or parent requirements into an agreed-to obligation for an entity to perform some function or possess some quality within specified constraints with acceptable risk.
\nRequirements are formal textual “shall” statements that communicate in a structured, natural language what an entity must do to realise the intent of the needs from which they were transformed. (Source: https://portal.incose.org/commerce/store?productId=INCOSE-GUIDEWRITINGREQ)", -- "A requirement statement is the result of a formal transformation of one or more needs or parent requirements into an agreed-to obligation for an entity to perform some function or possess some quality within specified constraints with acceptable risk.\n\nRequirements are formal textual “shall” statements that communicate in a structured, natural language what an entity must do to realise the intent of the needs from which they were transformed. (Source: https://portal.incose.org/commerce/store?productId=INCOSE-GUIDEWRITINGREQ)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:RISK--%3COF--A--HAZARD%3E\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ combination of the probability of occurrence of accident and the severity of that accident
\n
\n Note 1 to entry: In French, the term “risque” also denotes the potential source of harm, in English “hazard” (see 903-01-02) SPPRAMSS-4044 - Hazard .
\n [SOURCE: IEC 60050-903:2013, 903-01-07, modified — <of a hazard> has been added and "harm" has been replaced with "accident"] """, -- """ combination of the probability of occurrence of accident and the severity of that accident\n\n \n\n Note 1 to entry: In French, the term “risque” also denotes the potential source of harm, in English “hazard” (see 903-01-02) SPPRAMSS-4044 - Hazard .\n\n [SOURCE: IEC 60050-903:2013, 903-01-07, modified — "harm" has been replaced with "accident"] """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:RISK--ANALYSIS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "systematic use of available information to identify hazards and to estimate the risk
\n[Source: IEC 60050-903:2013, 903-01-08]", -- "systematic use of available information to identify hazards and to estimate the risk \n\n[Source: IEC 60050-903:2013, 903-01-08]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:RISK--ASSESSMENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "overall process comprising a risk analysis and a risk evaluation
\n[Source: IEC 60050-903:2013, 903-01-10]", -- "overall process comprising a risk analysis and a risk evaluation\n\n[Source: IEC 60050-903:2013, 903-01-10]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:RISK--EVALUATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "procedure based on the risk analysis to determine whether the tolerable risk has been achieved
\n[Source: IEC 60050-903:2013, 903-01-09]", -- "procedure based on the risk analysis to determine whether the tolerable risk has been achieved \n\n[Source: IEC 60050-903:2013, 903-01-09]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SAFE--STATE--\_821-12-49\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "condition which continues to preserve safety
\n
\n[SOURCE: IEC 62425:2007, 3.1.42]", -- "condition which continues to preserve safety \n\n \n\n[SOURCE: IEC 62425:2007, 3.1.42]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SAFE--STATE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "condition which continues to preserve safety
\n [Source: IEC 60050-821, 821-12-49]", -- "condition which continues to preserve safety\n\n [Source: IEC 60050-821, 821-12-49]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SAFE--STATE\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "condition which continues to preserve safety
\n [821-12-49][IEC 62425:2007, 3.1.42]", -- "condition which continues to preserve safety \n\n [821-12-49][IEC 62425:2007, 3.1.42]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SAFETY--INVARIANT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Safety requirement regarding conditions or properties that are critical for maintaining the safe operation of a system. It ensures that the system does not enter into a hazardous state or condition.
\nNote: By defining and monitoring an invariant, engineers and safety professionals can detect deviations from expected behavior and take corrective actions to prevent accidents or failures. Invariants are often used in formal methods, fault tree analysis, hazard analysis, and other safety assessment techniques to ensure the robustness and reliability of systems.", -- "Safety requirement regarding conditions or properties that are critical for maintaining the safe operation of a system. It ensures that the system does not enter into a hazardous state or condition.\n\nNote: By defining and monitoring an invariant, engineers and safety professionals can detect deviations from expected behavior and take corrective actions to prevent accidents or failures. Invariants are often used in formal methods, fault tree analysis, hazard analysis, and other safety assessment techniques to ensure the robustness and reliability of systems."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SEMP--REQUIREMENTS--TYPES\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Common Business Objectives (CBO): High level objectives defining the mandate of the System Pillar. They are derived by Tasks and domains. They are not created in the Tasks and Domains. They can be formulated freely;{comment:48}
\n\* Railway Requirements: are operational epics that formulate concrete visions and user stories for the business target picture of an operational area (like CCS or Energy). The can be freely formulated but should follow the writing patterns of epics and user stories. They shall be specific enough to be assigned to exactly one System Pillar Task;{comment:49}
\n\* Operational Requirements: Are precise requirements that the Operational Analysis shall fulfil. {comment:24}They include PRAMSS and other non functional requirements;
\n\* Functional System Requirements: {comment:7}they are system requirements that the System Analysis, Logical Architecture or Subsystem Architecture shall fulfil and specify the functions of the System;{comment:14}
\n\* Non-Functional System Requirements: Are precise {comment:15}requirements that the System Analysis, Logical Architecture or Subsystem Architecture{comment:8} shall fulfil. They include PRAMSS and other non functional requirements.{comment:19}
\n\* Application Conditions + SRAC definitions: are precise requirements that the {comment:25}environment of the System In Use shall fulfill. They include physical needs, skill levels of maintenance personal, temperatures of server rooms, engineering rules, etc. The SRAC{comment:58} are specific application conditions relevant to safety.", -- "\* Common Business Objectives (CBO): High level objectives defining the mandate of the System Pillar. They are derived by Tasks and domains. They are not created in the Tasks and Domains. They can be formulated freely;{comment:48}\n\n\* Railway Requirements: are operational epics that formulate concrete visions and user stories for the business target picture of an operational area (like CCS or Energy). The can be freely formulated but should follow the writing patterns of epics and user stories. They shall be specific enough to be assigned to exactly one System Pillar Task;{comment:49}\n\n\* Operational Requirements: Are precise requirements that the Operational Analysis shall fulfil. {comment:24}They include PRAMSS and other non functional requirements;\n\n\* Functional System Requirements: {comment:7}they are system requirements that the System Analysis, Logical Architecture or Subsystem Architecture shall fulfil and specify the functions of the System;{comment:14}\n\n\* Non-Functional System Requirements: Are precise {comment:15}requirements that the System Analysis, Logical Architecture or Subsystem Architecture{comment:8} shall fulfil. They include PRAMSS and other non functional requirements.{comment:19}\n\n\* Application Conditions + SRAC definitions: are precise requirements that the {comment:25}environment of the System In Use shall fulfill. They include physical needs, skill levels of maintenance personal, temperatures of server rooms, engineering rules, etc. The SRAC{comment:58} are specific application conditions relevant to safety."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SERA-CCS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "SERA-CCS{comment:3} Single European Railway Area - Command Control and Signaling:
\n
\n For the railway sector, EU policy focuses on the main goal of establishing a single European railway area, an EU-wide system of railway networks which would allow the expansion of the rail sector based on
\n
\n\* competition,
\n\* technical harmonization and
\n\* joint development of cross-border connections
\n[see Building the single European railway area].
\n
\n To this end, the EU aims to:
\n
\n\* open and restructure the rail market
\n\* increase competitiveness by creating a level playing field for companies
\n\* develop infrastructure to ensure interoperability
\n\* improve efficiency in infrastructure use and safety
\n\* ensure fair prices for consumers{comment:5}
\nSERA-CCS is understood to consist of those SERA objectives which are relevant for CCS (technical/operational target). It is defined by the following capabilities:
\n
\n\* Provides Independence between Infrastructure Managers (IM) and Railway Undertakings (RU) or Vehicle Owners (VO):
\n \* No track access restriction or extra costs for a compliant RU/VO fleet/services accessing a given IM infrastructure compliant with SERA-CCS.
\n \* Non discriminatory access to IM services and capacity allocation across SERA-CCS IMs for compliant RU/VO fleet/services.
\n\* Ensures interoperability, which is the ability to allow the safe and uninterrupted movement of trains that accomplish the specified levels of performance across different infrastructure networks (IMs) and the ability that an infrastructure network is able to interact with trains of different RUs/VOs.
\n\* Ensures Harmonized Operations, based on:
\n \* Radio-based CAB Signalling,
\n \* Automatic Train{comment:1} Operation (ATO),
\n \* Single licensing for Drivers and Signalers across SERA-CCS.
\n\* Provides Backwards Compatibility:
\n \* Compatibility is considered to be achieved for a particular combination of on-board and trackside when the on-board is able to run a normal service on that trackside. The expression “train is running a normal service” shall be understood as “a train not penalized because of a reduction of performance or safety”.
\n \* {comment:2}The full functionality of any old version shall always be available in any new version of SERA-CCS
\n\* Enables significantly lower life cycle costs having easy Upgradability, Scaleability, Adaptability and Changeability of Hardware and Software components/systems/subsystems
\n\* Consists of a Modular Architecture, both trackside and onboard:
\n \* Trackside and Onboard architecture are designed to allow for separate replacements/upgrades of individual sub-systems with reasonable integration effort.
\n\* Enables Interchangeability of hardware and software components/systems/subsystems across suppliers
\n \* The modular architecture shall also allow for situations where the replacement/upgrade of individual components from different suppliers is needed.
\n
\n{comment:4}", -- "SERA-CCS{comment:3} Single European Railway Area - Command Control and Signaling:\n\n \n\n For the railway sector, EU policy focuses on the main goal of establishing a single European railway area, an EU-wide system of railway networks which would allow the expansion of the rail sector based on\n\n \n\n\* competition,\n\n\* technical harmonization and\n\n\* joint development of cross-border connections\n\n[see Building the single European railway area].\n\n \n\n To this end, the EU aims to:\n\n \n\n\* open and restructure the rail market\n\n\* increase competitiveness by creating a level playing field for companies\n\n\* develop infrastructure to ensure interoperability\n\n\* improve efficiency in infrastructure use and safety\n\n\* ensure fair prices for consumers{comment:5}\n\nSERA-CCS is understood to consist of those SERA objectives which are relevant for CCS (technical/operational target). It is defined by the following capabilities:\n\n \n\n\* Provides Independence between Infrastructure Managers (IM) and Railway Undertakings (RU) or Vehicle Owners (VO):\n\n \* No track access restriction or extra costs for a compliant RU/VO fleet/services accessing a given IM infrastructure compliant with SERA-CCS.\n\n \* Non discriminatory access to IM services and capacity allocation across SERA-CCS IMs for compliant RU/VO fleet/services.\n\n\* Ensures interoperability, which is the ability to allow the safe and uninterrupted movement of trains that accomplish the specified levels of performance across different infrastructure networks (IMs) and the ability that an infrastructure network is able to interact with trains of different RUs/VOs.\n\n\* Ensures Harmonized Operations, based on:\n\n \* Radio-based CAB Signalling,\n\n \* Automatic Train{comment:1} Operation (ATO),\n\n \* Single licensing for Drivers and Signalers across SERA-CCS.\n\n\* Provides Backwards Compatibility:\n\n \* Compatibility is considered to be achieved for a particular combination of on-board and trackside when the on-board is able to run a normal service on that trackside. The expression “train is running a normal service” shall be understood as “a train not penalized because of a reduction of performance or safety”.\n\n \* {comment:2}The full functionality of any old version shall always be available in any new version of SERA-CCS\n\n\* Enables significantly lower life cycle costs having easy Upgradability, Scaleability, Adaptability and Changeability of Hardware and Software components/systems/subsystems\n\n\* Consists of a Modular Architecture, both trackside and onboard:\n\n \* Trackside and Onboard architecture are designed to allow for separate replacements/upgrades of individual sub-systems with reasonable integration effort.\n\n\* Enables Interchangeability of hardware and software components/systems/subsystems across suppliers\n\n \* The modular architecture shall also allow for situations where the replacement/upgrade of individual components from different suppliers is needed.\n\n\n\n{comment:4}"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SERIOUS--ACCIDENT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "‘serious accident’ means any train collision or derailment of trains resulting in the death of at least one person or serious injuries to five or more persons or extensive damage to rolling stock, the infrastructure or the environment, and any other accident with the same consequences which has an obvious impact on railway safety regulation or the management of safety; ‘extensive damage’ means damage that can be immediately assessed by the investigating body to cost at least EUR 2 million in total;
\n[SOURCE: SPPRAMSS-337 - [Directive (EU) 2016/798] Article 3 Definitions (12) ]", -- "‘serious accident’ means any train collision or derailment of trains resulting in the death of at least one person or serious injuries to five or more persons or extensive damage to rolling stock, the infrastructure or the environment, and any other accident with the same consequences which has an obvious impact on railway safety regulation or the management of safety; ‘extensive damage’ means damage that can be immediately assessed by the investigating body to cost at least EUR 2 million in total;\n\n[SOURCE: SPPRAMSS-337 - [Directive (EU) 2016/798] Article 3 Definitions (12) ]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SITUATION\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A logical combination of modes and states using logical operations (AND, OR, NOT) representing the superposition of modes
\n and states occurring simultaneously at a given moment.", -- "A logical combination of modes and states using logical operations (AND, OR, NOT) representing the superposition of modes \n\n and states occurring simultaneously at a given moment."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SPECIFICATION--TASK\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Result: specification for an artifact
\n\* Usually ends on CENELEC P4
\n\* Left side of the CENELEC “V”", -- "\* Result: specification for an artifact\n\n\* Usually ends on CENELEC P4\n\n\* Left side of the CENELEC “V”"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SRD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Single Rules Database
\n[SOURCE: ERA, https://www.era.europa.eu/domains/registers/srd\_en ]", -- "Single Rules Database\n\n[SOURCE: ERA, https://www.era.europa.eu/domains/registers/srd\_en ]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:STAKEHOLDER--NEEDS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "These needs are taken into account for the definition of the System of interest.
\n Note: In the Europe's Rail context, stakeholder needs are defined by valid input channels, which are
\n - Decided SP Common Business Objectives as the top-level of the requirement tree,
\n - Requirements proposed by any party and approved in the SP decision process according to SP governance.", -- "These needs are taken into account for the definition of the System of interest.\n\n Note: In the Europe's Rail context, stakeholder needs are defined by valid input channels, which are\n\n - Decided SP Common Business Objectives as the top-level of the requirement tree,\n\n - Requirements proposed by any party and approved in the SP decision process according to SP governance."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:STAKEHOLDER\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Someone who is entitled to express needs for the system of interest.
\nNote: In the Europe's Rail context, the list of stakeholders is defined by the members of the System Pillar Steering group and their delegates or speakers.", -- "Someone who is entitled to express needs for the system of interest.\n\nNote: In the Europe's Rail context, the list of stakeholders is defined by the members of the System Pillar Steering group and their delegates or speakers."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:STPA\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "System Theoretic Process Analysis
\n[Source: J3187:202305]", -- "System Theoretic Process Analysis\n\n[Source: J3187:202305]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SUB-SYSTEM--SOMETIMES--CALLED--\_BUILDING--BLOCK\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Sub-systems are along ARCADIA systems on System Level 5. Not to be confused with sub-systems in the TSI / interoperability directive. In the TSI / interoperability directive context a sub-system shall be regarded as a interoperability constituent
\n A sub-system is a part of a system, which is not split into smaller entities. It represents a leaf element in the hierarchy of systems-of-systems.
\n Physically speaking, a sub-system is either a piece of hardware plus software, or just a piece of software.
\n A sub-system is a source able unit of the CCS system, in particular:
\n\* a sub-system can be individually tendered to a supplier,
\n\* a sub-system can be built individually by a supplier,
\n\* a sub-system must be integrated into a system, which includes all necessary test, verification, certification and validation activities depending on the level of harmonisation.
\nThe harmonisation of the sub-system’s features is to be defined according to the requested level:
\n
\n\* Functional Apportionment,
\n\* Interoperability,
\n\* Exchangeability, or
\n\* Interchangeability.", -- "sub-systems are along ARCADIA systems with standard interfaces on System Level 5. Not to be confused with sub-systems in the TSI / interoperability directive. In the TSI / interoperability directive context a subssystem shall be regarded as a interoperability constituent\n\n A sub-system is a part of a system, which is not split into smaller entities : it represents a leaf element in the hierarchy of systems-of-systems.\n\n Physically speaking, a sub-system is either a piece of hardware + software, or just a piece of software.\n\n A sub-system is a sourceable unit of the CCS system. In particular : \n\n\* A sub-system can be individually tendered to a supplier\n\n\* A sub-system can be built individually by a supplier\n\n\* A sub-system must be integrated into a system, which includes all necessary test, verification, certification and validation activities depending on the level of standardisation.\n\nThe standardization of the sub-system’s features is to be defined according to the requested level: Functional apportionment / Interoperability / Exchangeability / Interchangeability."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SUBSYSTEM\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Subsystems are along ARCADIA systems with standard interfaces on System Level 5. Not to be confused with subsystems in the TSI / interoperability directive.
\n A subsystem is a part of a system, which is not refined any further during the specification task(The term subsystem is used in this document following the referenced architecting standards, it does not correspond to the subsystem definition as in the Interoperability Directive.). It represents a leaf element in the hierarchy of systems-of-systems. A subsystem is defined by the following characteristics:
\n\* For each subsystem exists a set of specification documents, that allows a supplier to build that subsystem, ideally without the need for further documentation.
\n\* The level of strictness of the specification can be variable:
\n \* Interoperable specification: Strict standardisation of all interface aspects that are needed for to systems to fulfil a defined set of operational capabilities together on runtime.
\n \* Interchangeable specification: Standardisation of all interface aspects that are needed to exchange one of the systems with the lowest reasonable integration effort.
\n \* Core standardisation specification: An interoperable standardisation that defines a guaranteed minimum of interoperability, but allows and gives room for specific and perhaps incompatible extensions that are only used, when all involved systems have them.
\n \* Guideline: A recommended specification that can be used as a whole, or can be used partly or changed.
\n\* A subsystem can be implemented as software only, as hardware only, as a mixture of both depending on the strictness of the specification (the specification can leave that aspect open).
\n\* Each subsystem can be individually tendered to a supplier
\n\* Each subsystem can be built individually by a supplier
\n\* Each subsystem must be integrated into a system, which includes all necessary test, verification, certification and validation activities.
\nSome non-functional requirements (e.g. weight constraints or physical dimension constraints) will be allocated to subsystems.", -- "Subsystems are along ARCADIA systems with standard interfaces on System Level 5. Not to be confused with subsystems in the TSI / interoperability directive.\n\n A subsystem is a part of a system, which is not refined any further during the specification task(The term subsystem is used in this document following the referenced architecting standards, it does not correspond to the subsystem definition as in the Interoperability Directive.). It represents a leaf element in the hierarchy of systems-of-systems. A subsystem is defined by the following characteristics:\n\n\* For each subsystem exists a set of specification documents, that allows a supplier to build that subsystem, ideally without the need for further documentation.\n\n\* The level of strictness of the specification can be variable:\n\n \* Interoperable specification: Strict standardisation of all interface aspects that are needed for to systems to fulfil a defined set of operational capabilities together on runtime.\n\n \* Interchangeable specification: Standardisation of all interface aspects that are needed to exchange one of the systems with the lowest reasonable integration effort.\n\n \* Core standardisation specification: An interoperable standardisation that defines a guaranteed minimum of interoperability, but allows and gives room for specific and perhaps incompatible extensions that are only used, when all involved systems have them.\n\n \* Guideline: A recommended specification that can be used as a whole, or can be used partly or changed.\n\n\* A subsystem can be implemented as software only, as hardware only, as a mixture of both depending on the strictness of the specification (the specification can leave that aspect open).\n\n\* Each subsystem can be individually tendered to a supplier\n\n\* Each subsystem can be built individually by a supplier\n\n\* Each subsystem must be integrated into a system, which includes all necessary test, verification, certification and validation activities.\n\nSome non-functional requirements (e.g. weight constraints or physical dimension constraints) will be allocated to subsystems."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SUC\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "System under Consideration
\n [SOURCE: SPPRAMSS-4697 - [EN IEC 62443-3-2:2020]]", -- "System under Consideration\n\n [SOURCE: SPPRAMSS-4697 - [IEC 62443-3-2:2020]]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SWITCH\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Switch", -- "a switch is a device that opens or closes electrical circuits."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SWITCH\_lexConcept\_4 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "a switch is a device that opens or closes electrical circuits.", -- "Electrical switch. Closes or opens a circuit."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--AND--INNOVATION--PROGRAMME--BOARD\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "System and Innovation Programme Board
\n advises the Executive Director on project and programme management of the JU including interaction between the two pillars as well as change management and conflicts, supported by the System Pillar Core Group", -- "System and Innovation Programme Board\n\n advises the Executive Director on project and programme management of the JU including interaction between the two pillars as well as change management and conflicts, supported by the System Pillar Core Group"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--DEVELOPMENT--LIFE--CYCLE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The System Development Life Cycle (SDLC) is a structured approach or methodology used in system engineering to guide the creation, deployment, and maintenance of systems. It provides a systematic framework for managing and controlling the entire process of building a system, from inception to retirement.
\n The SDLC provides a structured approach to system development, ensuring that all necessary activities are carried out in a logical and controlled manner. It helps in managing risks, controlling costs, and delivering high-quality systems that meet stakeholder expectations.", -- "The System Development Life Cycle (SDLC) is a structured approach or methodology used in system engineering to guide the creation, deployment, and maintenance of systems. It provides a systematic framework for managing and controlling the entire process of building a system, from inception to retirement.\n\n The SDLC provides a structured approach to system development, ensuring that all necessary activities are carried out in a logical and controlled manner. It helps in managing risks, controlling costs, and delivering high-quality systems that meet stakeholder expectations."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--LEVELS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The following figure shows the decomposition of a system of systems on one consistent example spanning 5 layers of refinement. Level 5 is the actual subsystem layer and is visually integrated into the bottom layer in the following figure to be able to show the relationship to logical components.
\n(image: 1-screenshot-20221229-202431.png)
\nFigure {caption:Figure}: System Level 1-5 combined view", -- "The following figure shows the decomposition of a system of systems on one consistent example spanning 5 layers of refinement. Level 5 is the actual subsystem layer and is visually integrated into the bottom layer in the following figure to be able to show the relationship to logical components. \n\n(image: 1-screenshot-20221229-202431.png)\n\nFigure {caption:Figure}: System Level 1-5 combined view"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--OF--SYSTEMS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "According to ISO 21839, a system of systems is set of systems that interact to provide a unique capability that none of the constituent systems can accomplish on its own.
\nA system of systems in our understanding at least comprises two constituent systems.
\nA system of systems itself can be nested as a constituent system in a larger system of systems, i.e. system of systems can span multiple levels recursively.", -- "According to ISO 21839, a system of systems is set of systems that interact to provide a unique capability that none of the constituent systems can accomplish on its own.\n\nA system of systems in our understanding at least comprises two constituent systems.\n\nA system of systems itself can be nested as a constituent system in a larger system of systems, i.e. system of systems can span multiple levels recursively."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--PILLAR--CORE--GROUP\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "System Pillar Core Group
\n Manages the common business objectives and deliverables from the Tasks", -- "System Pillar Core Group\n\n Manages the common business objectives and deliverables from the Tasks"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--PILLAR--DELIVERABLES--\_OUTPUT--DOCUMENTS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Deliverables are “output documents” that are automatically or manually created and exported from the engineering database (ALM) in the predefined structure of the SP deliverable templates
\n
\n List of deliverables per subsystem (System Level 5).
\n ORS is done for System Level 3, and then tailored for the subsystems (especially CONEMP processes are done only on System Level 5).
\n
\n System specific Operational Requirement Specification (ORS)
\n D2.1 As-Is analysis D2.2 Referred CBO and railway requirements D2.3 Problem analysis and derived process improvements D2.4 Application categories D2.5 Operational requirements (incl. non-functional and process requirements) D2.6 Operational entities and actor D2.7 Operational capabilities D2.8 PRAMSS targets, strategies and indicators D2.9 Operational processes D2.10 Operational hazards and risks D2.11 Rule books for all actors D2.12 The concept for the operational migrationFunctional Requirement Specification (FRS)
\n D3.1 System definition D3.2 Detailed system actor descriptions and roles D3.3 System capabilities D3.4 Functional chains and sequences per capability D3.5 Function specification D3.6 Functional hazards and risksSystem requirement Specification (SRS)
\n D4.1 Architecture of systems of the next level (if standardized) D4.2 Functional allocation to logical components D4.3 Physical architecture D4.4 Technical and physical hazards and risks D4.5 The technical migration strategy is defined D4.6 System requirements and interface specification, incl. legacy adpaters D4.7 Non-functional System requirementsApplication specification (ARS)
\n D5.1 Release and Implementation Configurations D5.2 Application conditions, CSM/CST, HSI Plan D5.3 Application/life cycle/usage guideline/rules D5.4 Engineering and maintenance guideline/rules D5.5 Validation and test specificationValidation and test specification (VRS)
\n D6.1 Model checking specification D6.2 Simulation/test environment specification D6.3 Simulation/test/validation cases and dataStandardisation and CCM documentation
\n D7.1 Standardisation packages and publications D7.2 External quality and experience monitoring for the standard is set up D7.2 External change management documentation", -- "Deliverables are “output documents” that are automatically or manually created and exported from the engineering database (ALM) in the predefined structure of the SP deliverable templates\n\n \n\n List of deliverables per subsystem (System Level 5).\n\n ORS is done for System Level 3, and then tailored for the subsystems (especially CONEMP processes are done only on System Level 5).\n\n \n\n System specific Operational Requirement Specification (ORS)\n\n D2.1 As-Is analysis D2.2 Referred CBO and railway requirements D2.3 Problem analysis and derived process improvements D2.4 Application categories D2.5 Operational requirements (incl. non-functional and process requirements) D2.6 Operational entities and actor D2.7 Operational capabilities D2.8 PRAMSS targets, strategies and indicators D2.9 Operational processes D2.10 Operational hazards and risks D2.11 Rule books for all actors D2.12 The concept for the operational migrationFunctional Requirement Specification (FRS)\n\n D3.1 System definition D3.2 Detailed system actor descriptions and roles D3.3 System capabilities D3.4 Functional chains and sequences per capability D3.5 Function specification D3.6 Functional hazards and risksSystem requirement Specification (SRS)\n\n D4.1 Architecture of systems of the next level (if standardized) D4.2 Functional allocation to logical components D4.3 Physical architecture D4.4 Technical and physical hazards and risks D4.5 The technical migration strategy is defined D4.6 System requirements and interface specification, incl. legacy adpaters D4.7 Non-functional System requirementsApplication specification (ARS)\n\n D5.1 Release and Implementation Configurations D5.2 Application conditions, CSM/CST, HSI Plan D5.3 Application/life cycle/usage guideline/rules D5.4 Engineering and maintenance guideline/rules D5.5 Validation and test specificationValidation and test specification (VRS)\n\n D6.1 Model checking specification D6.2 Simulation/test environment specification D6.3 Simulation/test/validation cases and dataStandardisation and CCM documentation\n\n D7.1 Standardisation packages and publications D7.2 External quality and experience monitoring for the standard is set up D7.2 External change management documentation"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--PILLAR--STEERING--GROUP\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "System Pillar Steering Group
\n monitoring the progress of the System Pillar", -- "System Pillar Steering Group\n\n monitoring the progress of the System Pillar"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--PILLAR--UNIT--CHAIRS--THE--SYSTEM--PILLAR--CORE--GROUP\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "System Pillar Unit
\n Chairs the System Pillar Core Group", -- "System Pillar Unit\n\n Chairs the System Pillar Core Group"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM--REQUIREMENTS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "define the constraints that affect the whole or a significant proportion of the system and include:
\n\* Physical Attributes:
\n \* System Style
\n \* System Size
\n \* System Weight
\n\* The 'RAMI' properties:
\n \* System Reliability
\n \* System Availability
\n \* System Maintainability
\n \* System Interoperability", -- "define the constraints that affect the whole or a significant proportion of the system and include: \n\n\* Physical Attributes:\n\n \* System Style\n\n \* System Size\n\n \* System Weight\n\n\* The 'RAMI' properties:\n\n \* System Reliability\n\n \* System Availability\n\n \* System Maintainability\n\n \* System Interoperability"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:SYSTEM\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "According to ISO 15288 a system is “a combination of interacting elements organized to achieve one or more stated purposes. “. In terms of this document, a system in black box view is furthermore defined by:
\n\* interfaces to actors outside the system, defining the system boundary
\n\* at least one function allocated to it
\n\* at least one system capability that the system delivers as a service to the actors
\nA system in white box view can be further refined into (exclusive or):
\n\* into a more granular systems, hence making itself to a system of systems
\n\* into subsystems on the lowest level of system of systems refinement
\nIn both cases, a system is a conceptual entity that aggregates the properties of its constituents but is not the element that defines the properties itself. A system is hence subject to the emerging properties of its constituents. Example: The system “onboard CCS” does not define the properties of this system itself, the properties emerge from the combined properties of the subsystems “ETCS core”, “Onboard localization” and “Onboard map” (examples only).
\nUsage context definitions of term „system“:
\n\* Constituent system: according to ISO 21839, a system that forms part of a system of systems
\n\* System of interest: according to ISO 21839, a system whose life cycle or properties are under consideration in a given context", -- "According to ISO 15288 a system is “a combination of interacting elements organized to achieve one or more stated purposes. “. In terms of this document, a system in black box view is furthermore defined by: \n\n\* interfaces to actors outside the system, defining the system boundary\n\n\* at least one function allocated to it\n\n\* at least one system capability that the system delivers as a service to the actors\n\nA system in white box view can be further refined into (exclusive or): \n\n\* into a more granular systems, hence making itself to a system of systems\n\n\* into subsystems on the lowest level of system of systems refinement\n\nIn both cases, a system is a conceptual entity that aggregates the properties of its constituents but is not the element that defines the properties itself. A system is hence subject to the emerging properties of its constituents. Example: The system “onboard CCS” does not define the properties of this system itself, the properties emerge from the combined properties of the subsystems “ETCS core”, “Onboard localization” and “Onboard map” (examples only).\n\nUsage context definitions of term „system“: \n\n\* Constituent system: according to ISO 21839, a system that forms part of a system of systems\n\n\* System of interest: according to ISO 21839, a system whose life cycle or properties are under consideration in a given context"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TAILORING--OF--REQUIREMENT--BREAKDOWN\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "SoS engineering can go wrong (quite often) if done in an overengineered and abstract way. If every artefact is created on every System Level (e.g. 5-7 levels and types or requirements that are broken down), the workload is artificially multiplied without need, and this policy is not creating a really good work split.
\nTo avoid this, every work process uses a tailoring approach which means
\n\* “Copy” (reference) things from a higher level, if possible like they are
\n\* Break up and translate through the levels that you really need, and not more.
\n\* Be sure that the right requirements are broken down to System Level 5 specification and to support concept definition in projects and manage the system, its performance and standards management.
\n\* Operational rulebooks include all that is needed.
\n\* Support design phases in managing requirement allocation, interfaces managements, testing planning. Engineering rules are complete and fulfilling the requirements
\n\* Support integration phase supporting data analysis, simulation process, anomaly resolution
\nPrograms can fail (cost, duration, time to market) because of too long and intense workout on System Level 1-4 (overengineered analytical architecting) without getting to real implementation specification and questions.
\nIf for example a high level requirement coming from the stakeholders can be addressed directly to System Level 5 without risk, this should be done. This is a (workflow) “tailoring” decision to be taken in the functional teams.
\n Wrong approach: Breakdown every work item on every level.
\n Right approaches:
\n\* Specific workflow tailoring, and
\n\* Generic workflow tailoring
\n“Specific (workflow) tailoring” (intelligent work assignment per work item)
\n = Single work items are only assigned on levels, where work is really needed
\n (image: 1-image1.png)
\nSecond example for specific workflow tailoring: Two different process break downs
\n\* Op. process “Emergency stop of a train”, splits up “untailored” into work assignments…
\n \* System level 1/2: Define generic roles of CCS and TMS in this case
\n \* System Level 3 TMS: Define TMS process for it
\n \* System Level 3 CCS: Define generic roles and basic information flow of Level 4 CCS Systems for it
\n \* ….
\n \* System Level 4/5 Traffic CS: ….
\n \* System Level 4/5 Train CS: …
\n \* …
\n\* Operational process “Reset object controller” shall be tailored into these work assignments
\n \* System Level 1/2: Skipped
\n \* System Level 3: Delegate interface coordination to Traffic CS / Trackside Asset CS
\n \* System Level 4/5 Traffic CS: Define process for the remote reset
\n \* System Level 4/5 Traffic CS: Define process for the local OC reset
\n \* System Level 4/5 Trackside Asset CS: Define process for the remote OC reset
\n \* System Level 4/5 Trackside Asset CS: Define process for the local OC reset
\nConsequence: With specific workflow tailoring, the analytical system levels are incomplete (see table above).
\n
\n “Generic (Workflow) Tailoring”
\n = Certain work item types are worked out only on certain layers
\nExample:
\nExamples: | CBO | Requirements | Operational Processes | Logical Components
\nSystem Level 1/2 | X | X | Business process |
\nSystem Level 3/4 | | X | X |
\nSystem Level 5 | | X | X | XCoordination of “tailoring” is the main role of all “functional teams” (see chapter later in the document).
\nA good tailoring fulfils the following requirements
\n\* Work is parallelized early and in a precise way (independent work)
\n\* No “dummy” work items just for the sake of symmetric work item break downs in the system of systems
\n\* The derivation process does not “block” everything because of sequential working dependencies
\n\* The level of details of the specification is always only as good as needed on System Level 1-4 (analytical clarification = work preparation) so that the teams working on System Level 5 know what to do.
\n\* A domain team on System Level 5 – the real specification level – can concentrate on its local work functions, engineering rules, and the collaboration with other domain teams concerning concrete interfaces
\n\* A task level architecture team (ARC, OD, MIG) just focusses on the role to provide all needed specifications (especially requirements, process definitions, functional allocation) that enables the work on System Level 5 Team - without already creating too much design
\n“Tailoring” is the work to dynamically assign work items to the system level that shall work on them. A tailoring process is executed by a functional team. The typical workflow for tailoring is
\n\* An “inbox” (automated workflow, state driven) for new/changed work items is automated (e.g. inbox for new requirements, or for a proposal of process design change)
\n\* Some work items can be assigned automatically by rules (generic tailoring)
\n\* Else: The responsible person (role owner in the functional team) checks (evtl. discusses) the new work item
\n\* The new work item is assigned to a system level and team.
\nThis happens to all types of inputs and changes.
\n Conclusions:
\n\* Do only the minimal needed work on System Level 1-4, needed to get good Level 5 specifications, rulebooks, etc.
\n\* Apply a generic tailoring to avoid that everybody is working on the same things
\n\* Design a workflow that assigns work items dynamically to the right level.", -- "SoS engineering can go wrong (quite often) if done in an overengineered and abstract way. If every artefact is created on every System Level (e.g. 5-7 levels and types or requirements that are broken down), the workload is artificially multiplied without need, and this policy is not creating a really good work split.\n\nTo avoid this, every work process uses a tailoring approach which means \n\n\* “Copy” (reference) things from a higher level, if possible like they are\n\n\* Break up and translate through the levels that you really need, and not more.\n\n\* Be sure that the right requirements are broken down to System Level 5 specification and to support concept definition in projects and manage the system, its performance and standards management.\n\n\* Operational rulebooks include all that is needed.\n\n\* Support design phases in managing requirement allocation, interfaces managements, testing planning. Engineering rules are complete and fulfilling the requirements\n\n\* Support integration phase supporting data analysis, simulation process, anomaly resolution\n\nPrograms can fail (cost, duration, time to market) because of too long and intense workout on System Level 1-4 (overengineered analytical architecting) without getting to real implementation specification and questions.\n\nIf for example a high level requirement coming from the stakeholders can be addressed directly to System Level 5 without risk, this should be done. This is a (workflow) “tailoring” decision to be taken in the functional teams.\n\n Wrong approach: Breakdown every work item on every level. \n\n Right approaches: \n\n\* Specific workflow tailoring, and\n\n\* Generic workflow tailoring\n\n“Specific (workflow) tailoring” (intelligent work assignment per work item)\n\n = Single work items are only assigned on levels, where work is really needed\n\n (image: 1-image1.png) \n\nSecond example for specific workflow tailoring: Two different process break downs \n\n\* Op. process “Emergency stop of a train”, splits up “untailored” into work assignments…\n\n \* System level 1/2: Define generic roles of CCS and TMS in this case\n\n \* System Level 3 TMS: Define TMS process for it\n\n \* System Level 3 CCS: Define generic roles and basic information flow of Level 4 CCS Systems for it\n\n \* ….\n\n \* System Level 4/5 Traffic CS: ….\n\n \* System Level 4/5 Train CS: …\n\n \* …\n\n\* Operational process “Reset object controller” shall be tailored into these work assignments\n\n \* System Level 1/2: Skipped\n\n \* System Level 3: Delegate interface coordination to Traffic CS / Trackside Asset CS\n\n \* System Level 4/5 Traffic CS: Define process for the remote reset\n\n \* System Level 4/5 Traffic CS: Define process for the local OC reset\n\n \* System Level 4/5 Trackside Asset CS: Define process for the remote OC reset\n\n \* System Level 4/5 Trackside Asset CS: Define process for the local OC reset\n\nConsequence: With specific workflow tailoring, the analytical system levels are incomplete (see table above).\n\n \n\n “Generic (Workflow) Tailoring”\n\n = Certain work item types are worked out only on certain layers\n\nExample: \n\nExamples: | CBO | Requirements | Operational Processes | Logical Components\n\nSystem Level 1/2 | X | X | Business process | \n\nSystem Level 3/4 | | X | X | \n\nSystem Level 5 | | X | X | XCoordination of “tailoring” is the main role of all “functional teams” (see chapter later in the document).\n\nA good tailoring fulfils the following requirements \n\n\* Work is parallelized early and in a precise way (independent work)\n\n\* No “dummy” work items just for the sake of symmetric work item break downs in the system of systems\n\n\* The derivation process does not “block” everything because of sequential working dependencies\n\n\* The level of details of the specification is always only as good as needed on System Level 1-4 (analytical clarification = work preparation) so that the teams working on System Level 5 know what to do.\n\n\* A domain team on System Level 5 – the real specification level – can concentrate on its local work functions, engineering rules, and the collaboration with other domain teams concerning concrete interfaces\n\n\* A task level architecture team (ARC, OD, MIG) just focusses on the role to provide all needed specifications (especially requirements, process definitions, functional allocation) that enables the work on System Level 5 Team - without already creating too much design\n\n“Tailoring” is the work to dynamically assign work items to the system level that shall work on them. A tailoring process is executed by a functional team. The typical workflow for tailoring is \n\n\* An “inbox” (automated workflow, state driven) for new/changed work items is automated (e.g. inbox for new requirements, or for a proposal of process design change)\n\n\* Some work items can be assigned automatically by rules (generic tailoring)\n\n\* Else: The responsible person (role owner in the functional team) checks (evtl. discusses) the new work item\n\n\* The new work item is assigned to a system level and team.\n\nThis happens to all types of inputs and changes.\n\n Conclusions: \n\n\* Do only the minimal needed work on System Level 1-4, needed to get good Level 5 specifications, rulebooks, etc.\n\n\* Apply a generic tailoring to avoid that everybody is working on the same things\n\n\* Design a workflow that assigns work items dynamically to the right level."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TARGET--PICTURE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The following are considered minimum characteristics (baseline) for the targeted solution:
\n\* Operational processes directly related to the railway traffic commercial production are harmonised. This includes ETCS related processes, processes for securing and releasing railway lines/areas to/from maintenance and commissioning, operating through and recovering from degraded (non-regular) operations, reacting to and managing emergency situations (from operational perspective).
\n\* Railway system operates ETCS L2, without signals.
\n\* Railway system operates EULYNX type equipment.", -- "The following are considered minimum characteristics (baseline) for the targeted solution:\n\n\* Operational processes directly related to the railway traffic commercial production are harmonised. This includes ETCS related processes, processes for securing and releasing railway lines/areas to/from maintenance and commissioning, operating through and recovering from degraded (non-regular) operations, reacting to and managing emergency situations (from operational perspective).\n\n\* Railway system operates ETCS L2, without signals.\n\n\* Railway system operates EULYNX type equipment."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TECHNICAL--DELAY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "delay incurred in performing auxiliary technical actions associated with, but not part of, the maintenance action
\nEXAMPLE Rendering the equipment safe (such as immobilising, cooling, isolation and grounding).
\n[SOURCE: IEC 60050-192:2015,192-07-15]", -- "delay incurred in performing auxiliary technical actions associated with, but not part of, the maintenance action \n\nEXAMPLE Rendering the equipment safe (such as immobilising, cooling, isolation and grounding).\n\n[SOURCE: IEC 60050-192:2015,192-07-15]"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TEMPORARY--SHUNTING--AREA--TSHA--OR-DEF-160--TEMPORARY--SHUNTING--AREA--OR-DEF-161--DEF\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Temporary Shunting Area (TShA)
\nOR.DEF.160 Temporary shunting area
\n OR.DEF.161 DEFINITION A temporary shunting area is an interlocked area{comment:15} temporarily set up to allow shunting operations. A temporary shunting area is always under the responsibility of a Shunting area manager.", -- "Temporary Shunting Area (TShA)\n\nOR.DEF.160 Temporary shunting area\n\n OR.DEF.161 DEFINITION A temporary shunting area is an interlocked area{comment:15} temporarily set up to allow shunting operations. A temporary shunting area is always under the responsibility of a Shunting area manager."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TERM--STATUS--DESCRIPTION--OPERATING--STATE--DRAFT--THE--OPERATING--STATE--DESCRIBES--THE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ Term | Status | Description
\nOperating state | draft | The operating state describes the current state of production: \* how trains move\* where trains are\* which route is set in control system\* status of assetsThis information is based on inputs from control systems.
\nCapacity | draft | The total overall capability of the infrastructure that can be utilised by traffic or by maintenance. Capacity may apply to specific geographic sectors like stations or lines. Capacity usage must be requested, which results in capacity demand.
\nCapacity Restriction | draft | Capacity restriction is a temporary full or partial unavailability of network infrastructure due to construction works, maintenance, inspection works or due to environmental influences and disruptions.
\nCapacity Planning | draft | Capacity Planning is an instrument to determining the total theoretically available capacity supply (max. number of journeys per direction) and compare it to demand. Capacity can be requested from long term to short term (ad hoc slot ordering). Capacity planning supports the planning along all-time horizons (strategic to short term).
\nCapacity Plan | draft | The capacity plan comprises any planned capacity usage (traffic and construction works) at any point in time during the planning period. The aim is a consistent and conflict free capacity plan.
\nCapacity Production | draft | Capacity production is the implementation of the operational plan. Capacity production begins when the train starts its scheduled mission and ends with when it ends.
\nConflict | draft | A conflict is any difference between the operational plan and the forecast, regardless of any actual need for intervention to resolve it.
\nDispatching | draft | Dispatching is the sum of actions intended to make modifications the operational plan.
\nDeviation | draft | A deviation is any difference between the operational plan and the actual state of traffic; unlike a conflict, deviations may only be mitigated but not solved.
\nDecision | draft | A decision is a single action intended to make a modification the operational plan.
\nForecast | draft | The forecast displays the future state of traffic. It based on the current operational state. The forecast projects the current operational plan into the future. A forecast for each train run under consideration of mutual train influence is processed (train run time, section run time, minimal section run time, detailed topology, headways and train sequence, circulation, connection, etc.).
\nOperational Plan | draft | Analogous to the capacity plan, the operational plan supplies the train and traffic control and all other components with the operational train data. The operational plan is fed from the active timetable buffer and contains all trains (from the timetable buffer) that are currently in their operational time window. The operational plan is the result of various influences (decisions from dispatching, map data, etc.). It is updated every single minute.
\nMaster Data | draft | Master data represents "data about the business entities that provide context for business transactions". """, -- """ Term | Status | Description\n\nOperating state | draft | The operating state describes the current state of production: \* how trains move\* where trains are\* which route is set in control system\* status of assetsThis information is based on inputs from control systems.\n\nCapacity | draft | The total overall capability of the infrastructure that can be utilised by traffic or by maintenance. Capacity may apply to specific geographic sectors like stations or lines. Capacity usage must be requested, which results in capacity demand. \n\nCapacity Restriction | draft | Capacity restriction is a temporary full or partial unavailability of network infrastructure due to construction works, maintenance, inspection works or due to environmental influences and disruptions.\n\nCapacity Planning | draft | Capacity Planning is an instrument to determining the total theoretically available capacity supply (max. number of journeys per direction) and compare it to demand. Capacity can be requested from long term to short term (ad hoc slot ordering). Capacity planning supports the planning along all-time horizons (strategic to short term).\n\nCapacity Plan | draft | The capacity plan comprises any planned capacity usage (traffic and construction works) at any point in time during the planning period. The aim is a consistent and conflict free capacity plan.\n\nCapacity Production | draft | Capacity production is the implementation of the operational plan. Capacity production begins when the train starts its scheduled mission and ends with when it ends.\n\nConflict | draft | A conflict is any difference between the operational plan and the forecast, regardless of any actual need for intervention to resolve it.\n\nDispatching | draft | Dispatching is the sum of actions intended to make modifications the operational plan.\n\nDeviation | draft | A deviation is any difference between the operational plan and the actual state of traffic; unlike a conflict, deviations may only be mitigated but not solved.\n\nDecision | draft | A decision is a single action intended to make a modification the operational plan.\n\nForecast | draft | The forecast displays the future state of traffic. It based on the current operational state. The forecast projects the current operational plan into the future. A forecast for each train run under consideration of mutual train influence is processed (train run time, section run time, minimal section run time, detailed topology, headways and train sequence, circulation, connection, etc.).\n\nOperational Plan | draft | Analogous to the capacity plan, the operational plan supplies the train and traffic control and all other components with the operational train data. The operational plan is fed from the active timetable buffer and contains all trains (from the timetable buffer) that are currently in their operational time window. The operational plan is the result of various influences (decisions from dispatching, map data, etc.). It is updated every single minute.\n\nMaster Data | draft | Master data represents "data about the business entities that provide context for business transactions". """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TESTABILITY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A sub-system that is not designed for testability will not be ready to show that it fulfils the requirements needed by the overall system. Testability is not an attribute of the sub-system/module itself but has to be designed into architecture and interfaces.", -- "A sub-system that is not designed for testability will not be ready to show that it fulfills the requirements needed by the overall system. Testability is not an attribute of the sub-system/module itself but has to be designed into architecture and interfaces."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:THE--\_TRACE\_--FOR--A--WORK--ITEM--CHAIN\_TREE\_GRAPH\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* A Trace is a chain of all linked work items (a “directed graph” of nodes and links), that resolve all dependencies of a single work item
\n\* Example trace (in this case a “simple tree of nodes”, but can also be more complex):
\n \* A common business objective is solved by
\n \* a list of operational capabilities and requirements which are implemented in
\n \* a list of operational processes which need
\n \* a list of system capabilities and functions which are allocated in
\n \* logical components that are implemented in
\n \* physical subsystems which have
\n \* a list of precisely defined interfaces and
\n \* engineering rules..
\n\* A trace is valid if all derivations are complete and correct.
\n\* A trace is complete if all dependencies are resolved by specified and verified work items (down to the implementation of detailed operational rulebooks and system handbooks, engineering rules and FFFiS interface specifications)
\n\* The overall specification is complete when all objectives, operational capabilities, and operational requirements are resolved by complete traces.
\n\* Example of the trace of one requirement (red node) down to functions, processes and components (other colours)
\n
\n
\n (image: 1-Grafik\_1780564629.png)", -- "\* A Trace is a chain of all linked work items (a “directed graph” of nodes and links), that resolve all dependencies of a single work item\n\n\* Example trace (in this case a “simple tree of nodes”, but can also be more complex):\n\n \* A common business objective is solved by\n\n \* a list of operational capabilities and requirements which are implemented in\n\n \* a list of operational processes which need\n\n \* a list of system capabilities and functions which are allocated in\n\n \* logical components that are implemented in\n\n \* physical subsystems which have\n\n \* a list of precisely defined interfaces and\n\n \* engineering rules..\n\n\* A trace is valid if all derivations are complete and correct.\n\n\* A trace is complete if all dependencies are resolved by specified and verified work items (down to the implementation of detailed operational rulebooks and system handbooks, engineering rules and FFFiS interface specifications)\n\n\* The overall specification is complete when all objectives, operational capabilities, and operational requirements are resolved by complete traces.\n\n\* Example of the trace of one requirement (red node) down to functions, processes and components (other colours)\n\n\n\n \n\n (image: 1-Grafik\_1780564629.png)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TLS\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "Transport Layer Security
\n (source: SPPRAMSS-1705 - [UNSIG Subset-146] )", -- "Transport Layer Security\n\n (source: SPPRAMSS-1705 - [UNSIG Subset-146] )"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TRACK--FOOTPRINT\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "The track Footprint of an object is defined for a specific track. It is the perpendicular projection of this object's ground footprint on the track center line.
\nNotes: The definition corresponds to UNISIG's occupancy, without consideration fix-block / moving block. This definition is weak for intersections of two vehicles near points, where the width of vehicles has to be considered (e.g. with a fouling point indicator near the track) and for level crossings, because the road users are not associated with the rail track.", -- "The track Footprint of an object is defined for a specific track. It is the perpendicular projection of this object's ground footprint on the track center line.\n\nNotes: The definition corresponds to UNISIG's occupancy, without consideration fix-block / moving block. This definition is weak for intersections of two vehicles near points, where the width of vehicles has to be considered (e.g. with a fouling point indicator near the track) and for level crossings, because the road users are not associated with the rail track."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TRADE-SPACE--FACTOR\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ "trade-space factors" represent the different factors/characteristics defining the solution space/alternatives.
\nFrom this trade-off playspace containing set of good/acceptable solutions, we will need to identify the optimal one (i.e. "best" trade-off). """, -- """ "trade-space factors" represent the different factors/characteristics defining the solution space/alternatives.\n\nFrom this trade-off playspace containing set of good/acceptable solutions, we will need to identify the optimal one (i.e. "best" trade-off). """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:TRAIN-CENTRIC--TRACK--OCCUPANCY\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ With train-centric approach the trackside focus is on representing a railway vehicle with an extent in a topology rather than on representing anonymous occupancy states of fixed sizes reported by TTD systems as in today's conventional block-centric signaling systems. In other words, the term "train-centric" refers to a "train-oriented" view of the trackside system about the track occupation caused by a railway vehicle, e.g., an ETCS-equipped train. This view is achieved by fusion of localisation information from both train and trackside.
\n
\n A train-centric track occupancy determination is based primarily on localisation information received from the railway vehicle, e.g., from trains sent via ETCS Train Position Reports. The trackside system will also take into account additional trackside localisation information if available such as Trackside Train Detection (TTD) inputs, for example to:
\n\* Adjust the trackside view of track occupancy by train, based on clear TTD sections at the front or rear of the train
\n\* Detect movement of non-communicating railway vehicles, e.g., trains/wagons not equipped with ETCS and trains equipped with ETCS that have lost communication
\n\* Handle degraded situations, such as loss of train integrity. """, -- """ With train-centric approach the trackside focus is on representing a railway vehicle with an extent in a topology rather than on representing anonymous occupancy states of fixed sizes reported by TTD systems as in today's conventional block-centric signaling systems. In other words, the term "train-centric" refers to a "train-oriented" view of the trackside system about the track occupation caused by a railway vehicle, e.g., an ETCS-equipped train. This view is achieved by fusion of localisation information from both train and trackside.\n\n\n\n A train-centric track occupancy determination is based primarily on localisation information received from the railway vehicle, e.g., from trains sent via ETCS Train Position Reports. The trackside system will also take into account additional trackside localisation information if available such as Trackside Train Detection (TTD) inputs, for example to:\n\n\* Adjust the trackside view of track occupancy by train, based on clear TTD sections at the front or rear of the train\n\n\* Detect movement of non-communicating railway vehicles, e.g., trains/wagons not equipped with ETCS and trains equipped with ETCS that have lost communication\n\n\* Handle degraded situations, such as loss of train integrity. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:UPLINKING\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ """ To handle deviated new designs or new/changed external inputs the whole “engineering” has to be understood as
\n\* Work items with type, versions, and state
\n\* Their links with versions, link roles and state
\nThe engineering database is in the end a “directed graph” with “nodes” (work items) and links (graph just indicated in the picture):{comment:13}
\n (image: 3-image3.png)
\nThe basic principle for continuous integration using an ALM+MBSE system is based on the mathematical process of “adding a directed graph to another directed graph which can overlap in certain nodes”. This is called “Uplinking” in the System Pillar.
\n Existing graph (e.g. ALM master database) New graph fragment (external contribution)
\n (image: 1-screenshot-20221229-210015.png)
\n
\n“Merging these graphs” ="uplinking" (of conceptual, requirement and modelling work items) is the most feared job to do in system engineering when done manually. A strong system support is necessary. This merge can happen when
\n\* A daughter-system graph is “uplinked” to the mother-system (master) graph
\n\* An external contribution is integrated (perhaps again and again)
\n\* An external editor was used to edit elements in the mater database (e.g. ALM= master database, MBSE-Tool = editor for a part of the work items)
\n\* A bottom-up process is faster (domain team) then the top-down process (Level 3)
\n\* The architecture teams is faster concerning functional design then the operational design team with the process design. Or both are fast then the requirement feedback from the migration team is coming.
\n\* Etc.
\nA lot of isolated (time, team, engineering tool) asynchronous designs will occur in the System Pillar. They lead to these synchronization needs:
\n
\n (image: 2-screenshot-20221229-210015.png)
\n
\n
\nThis does not mean, that “synchronization” is the standard choice to integrate work. The preferred choice is top-down design and working in the same system whenever possible. But synchronization will anyway be needed very often.
\nThe sync process of these integration steps is the same. All integration work (conceptual level, system models, requirements, etc.) is about this merging process. The following things can happen:
\n\* A new node (work item) is added and linked to 0-n{comment:14} old nodes
\n\* An old node is changed or deleted or relinked in another way
\nThe indirect effect of such a change/add process can be
\n\* In a “trace” (chain of work items) links can get{comment:15} “suspect” if earlier links or nodes were changed (through partially-automated impact analysis)
\n\* New workflows are triggered by new work item states. If a new node is created with the state “draft”, it is because of this assigned by a tailoring process in a certain way.
\nTo add the new fragment{comment:16} to the exiting master database (can only be done efficiently with automated ALM systems) the following work steps are executed
\n\* Convert contribution to graph structure (work items/links), check consistency
\n\* Analyse and assess new contribution
\n\* Perhaps adapt/change contribution before synchronisation
\n\* Create difference to master database automatically
\n\* Assess difference and indirect impacts
\n\* Replace/link fragment in database (fully or partially) or revoke it
\n\* Check and analyse the new state of the master database
\nSimplification of synchronization
\n\* Assigned designated areas to be designed only in certain teams/tools/places (only connecting links to be synchronized)
\n\* Check separated designs together frequently to avoid large deviations
\nHow often synchronize/integrate?
\n\* Best: Directly work on the master model (possible with ALM, performance problems with MBSE tools)
\n\* Second best: Do the sync/integration very frequently (daily, weekly, or monthly)
\n\* Worst: do it once in 6 months --> Risk of unsolvable incompatibility and lost work is increased
\nConclusion:
\n\* We need a very efficient and sophisticated tool support for the synchronization/integration process
\n\* The integration/synchronization process will create large efforts in the Modelling Service Team
\nThe concrete rules, when and how “uplinking” is done, are defined in chapter 20. """, -- """ To handle deviated new designs or new/changed external inputs the whole “engineering” has to be understood as \n\n\* Work items with type, versions, and state\n\n\* Their links with versions, link roles and state\n\nThe engineering database is in the end a “directed graph” with “nodes” (work items) and links (graph just indicated in the picture):{comment:13}\n\n (image: 3-image3.png) \n\nThe basic principle for continuous integration using an ALM+MBSE system is based on the mathematical process of “adding a directed graph to another directed graph which can overlap in certain nodes”. This is called “Uplinking” in the System Pillar.\n\n Existing graph (e.g. ALM master database) New graph fragment (external contribution)\n\n (image: 1-screenshot-20221229-210015.png) \n\n \n\n“Merging these graphs” ="uplinking" (of conceptual, requirement and modelling work items) is the most feared job to do in system engineering when done manually. A strong system support is necessary. This merge can happen when \n\n\* A daughter-system graph is “uplinked” to the mother-system (master) graph\n\n\* An external contribution is integrated (perhaps again and again)\n\n\* An external editor was used to edit elements in the mater database (e.g. ALM= master database, MBSE-Tool = editor for a part of the work items)\n\n\* A bottom-up process is faster (domain team) then the top-down process (Level 3)\n\n\* The architecture teams is faster concerning functional design then the operational design team with the process design. Or both are fast then the requirement feedback from the migration team is coming.\n\n\* Etc.\n\nA lot of isolated (time, team, engineering tool) asynchronous designs will occur in the System Pillar. They lead to these synchronization needs:\n\n \n\n (image: 2-screenshot-20221229-210015.png) \n\n \n\n \n\nThis does not mean, that “synchronization” is the standard choice to integrate work. The preferred choice is top-down design and working in the same system whenever possible. But synchronization will anyway be needed very often.\n\nThe sync process of these integration steps is the same. All integration work (conceptual level, system models, requirements, etc.) is about this merging process. The following things can happen: \n\n\* A new node (work item) is added and linked to 0-n{comment:14} old nodes\n\n\* An old node is changed or deleted or relinked in another way\n\nThe indirect effect of such a change/add process can be \n\n\* In a “trace” (chain of work items) links can get{comment:15} “suspect” if earlier links or nodes were changed (through partially-automated impact analysis)\n\n\* New workflows are triggered by new work item states. If a new node is created with the state “draft”, it is because of this assigned by a tailoring process in a certain way.\n\nTo add the new fragment{comment:16} to the exiting master database (can only be done efficiently with automated ALM systems) the following work steps are executed \n\n\* Convert contribution to graph structure (work items/links), check consistency\n\n\* Analyse and assess new contribution\n\n\* Perhaps adapt/change contribution before synchronisation\n\n\* Create difference to master database automatically\n\n\* Assess difference and indirect impacts\n\n\* Replace/link fragment in database (fully or partially) or revoke it\n\n\* Check and analyse the new state of the master database\n\nSimplification of synchronization \n\n\* Assigned designated areas to be designed only in certain teams/tools/places (only connecting links to be synchronized)\n\n\* Check separated designs together frequently to avoid large deviations\n\nHow often synchronize/integrate? \n\n\* Best: Directly work on the master model (possible with ALM, performance problems with MBSE tools)\n\n\* Second best: Do the sync/integration very frequently (daily, weekly, or monthly)\n\n\* Worst: do it once in 6 months --> Risk of unsolvable incompatibility and lost work is increased\n\nConclusion: \n\n\* We need a very efficient and sophisticated tool support for the synchronization/integration process\n\n\* The integration/synchronization process will create large efforts in the Modelling Service Team\n\nThe concrete rules, when and how “uplinking” is done, are defined in chapter 20. """

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:WORK--ITEM--EDITOR\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* The basic system and workflow management tool is the ALM system (currently Polarion).
\n\* The ALM offers different frontend types to create/edit/change work items (document editors, database frontends, graphical design, etc.)
\n\* Additional editors (like graphical modelling tools for ARCADIA and SysML) can be used to edit elements of the engineering database in the ALM System (synchronized data) as far as a synchronisation software is available (see chapter data management)", -- "\* The basic system and workflow management tool is the ALM system (currently Polarion).\n\n\* The ALM offers different frontend types to create/edit/change work items (document editors, database frontends, graphical design, etc.)\n\n\* Additional editors (like graphical modelling tools for ARCADIA and SysML) can be used to edit elements of the engineering database in the ALM System (synchronized data) as far as a synchronisation software is available (see chapter data management)"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:WORK--ITEM\_lexConcept\_2 modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* ork items examples: “A requirement”, or “a function”, or “a concept aspect”
\n\* Work items are objects in the “engineering database” (ALM) that represent the result of a design step.
\n\* The structure of the work items is defined by the SEMP process definition documents (overview in the maps), that also make use of modelling standards like ARCADIA or SysML
\n\* The master-engineering database is the ALM System (currently Polarion) which contains all work items and their links.", -- "\* ork items examples: “A requirement”, or “a function”, or “a concept aspect”\n\n\* Work items are objects in the “engineering database” (ALM) that represent the result of a design step.\n\n\* The structure of the work items is defined by the SEMP process definition documents (overview in the maps), that also make use of modelling standards like ARCADIA or SysML\n\n\* The master-engineering database is the ALM System (currently Polarion) which contains all work items and their links."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:WORKFLOW--AND--WORKFLOW--RULES\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* A workflow describes the rules (processes) how a “trace” shall be created step by step.
\n\* A workflow includes a set of rules that define, how
\n \* work items must be broken down and linked (link rules)
\n \* work item types must be resolved
\n \* certain work item types are assigned to certain teams in the organisation
\n\* Every work item, that is not compliant to a workflow is marked as invalid
\n\* There are optional workflow steps in a workflow. E.g. a requirement can be directly resolved by multiple functions (that are always explained with a rational attribute) without having a “solution concept” work item, which is optional.", -- "\* A workflow describes the rules (processes) how a “trace” shall be created step by step.\n\n\* A workflow includes a set of rules that define, how\n\n \* work items must be broken down and linked (link rules)\n\n \* work item types must be resolved\n\n \* certain work item types are assigned to certain teams in the organisation\n\n\* Every work item, that is not compliant to a workflow is marked as invalid\n\n\* There are optional workflow steps in a workflow. E.g. a requirement can be directly resolved by multiple functions (that are always explained with a rational attribute) without having a “solution concept” work item, which is optional."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:WORKFLOW--PRIORITISATION--STRATEGY--TO--BE--DECIDED--PER--AREA\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "In general, there are two directions for prioritisation, which are “horizontal” or “vertical”
\n\* Horizontal prioritisation means for example “which of all operational capabilities do we break down first down to their operational processes (just the next derivation step per capability)”
\n\* Vertical prioritisation means for example “which operational capability do we resolve down to the physical architecture before we start to resolve the next
\n\* capability”
\n(image: 1-Grafik\_1.png)
\n\* Both prioritisation methods are mixed to reach two targets in parallel: a) Use the capacity of all teams in parallel with different roles (vertical prioritisation), b) get a more and more full picture top-down (horizontal prioritisation).
\n\* A “mix” can for example be: “At first only all operational capabilities for standard train movements (no shunting etc.) and only the processes related to ETCS Level R without ATO are traced down to the physical architecture in the first step”.
\nFor the prioritisation these aspects are important:
\n\* As much as possible early usable results shall be created (year by year).
\n \* Example 1 (“bottom-up”): All traces, that lead to the EULYNX interfaces shall be resolved and approved fast. EULYNX interfaces should be standardized soon.
\n \* This means to import and resolve all traces, and to connect all interfaces to the overall architecture “upwards” (connecting workflow)
\n \* Example 2 (“top-down”): All traces for DAC shall be resolved fast, but there are not so much complete (resolved) traces to import in terms of processes and architecture. All operational capabilities and operational requirements for DAC need to be created and broken down and resolved down to the implementation (=vertical prioritisation).
\n\* Important decisions shall be decided early as a basis for all work
\n \* Example 3: (“horizontal prioritisation”): The process standardisation scope and depth shall be decided as early as possible. This means to create all operational capabilities, derive them town to operational processes, and assess every process impact on the later grade of standardisation of products.
\n \* Example 4 (“horizontal + vertical prioritisation”): The functional allocation and ambition in the logical architecture shall be decided as early as possible. For this, all operational capabilities need to be derived down to the logical architecture, and every logical component needs to be designed concerning the external dependencies that it has for other logical components (e.g. delegation of safety requirements from one to other components get only visible with the deep dive in the functional design).", -- "In general, there are two directions for prioritisation, which are “horizontal” or “vertical” \n\n\* Horizontal prioritisation means for example “which of all operational capabilities do we break down first down to their operational processes (just the next derivation step per capability)”\n\n\* Vertical prioritisation means for example “which operational capability do we resolve down to the physical architecture before we start to resolve the next\n\n\* capability”\n\n(image: 1-Grafik\_1.png) \n\n\* Both prioritisation methods are mixed to reach two targets in parallel: a) Use the capacity of all teams in parallel with different roles (vertical prioritisation), b) get a more and more full picture top-down (horizontal prioritisation).\n\n\* A “mix” can for example be: “At first only all operational capabilities for standard train movements (no shunting etc.) and only the processes related to ETCS Level R without ATO are traced down to the physical architecture in the first step”.\n\nFor the prioritisation these aspects are important: \n\n\* As much as possible early usable results shall be created (year by year).\n\n \* Example 1 (“bottom-up”): All traces, that lead to the EULYNX interfaces shall be resolved and approved fast. EULYNX interfaces should be standardized soon.\n\n \* This means to import and resolve all traces, and to connect all interfaces to the overall architecture “upwards” (connecting workflow)\n\n \* Example 2 (“top-down”): All traces for DAC shall be resolved fast, but there are not so much complete (resolved) traces to import in terms of processes and architecture. All operational capabilities and operational requirements for DAC need to be created and broken down and resolved down to the implementation (=vertical prioritisation).\n\n\* Important decisions shall be decided early as a basis for all work\n\n \* Example 3: (“horizontal prioritisation”): The process standardisation scope and depth shall be decided as early as possible. This means to create all operational capabilities, derive them town to operational processes, and assess every process impact on the later grade of standardisation of products.\n\n \* Example 4 (“horizontal + vertical prioritisation”): The functional allocation and ambition in the logical architecture shall be decided as early as possible. For this, all operational capabilities need to be derived down to the logical architecture, and every logical component needs to be designed concerning the external dependencies that it has for other logical components (e.g. delegation of safety requirements from one to other components get only visible with the deep dive in the functional design)."

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:WORKFLOW--STEP--ON--STEP--IN--A--WORKITEM--TRACE\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "A workflow step includes some of the following actions
\n\* “Create a trace”: The work item on the “top/start” of a trace is created based on external decisions. These are typically
\n \* Steering group decisions about system capabilities, common business objectives, improvement targets or problem descriptions, operational requirements, new safety requirements, etc. (see chapter below).
\n \* Technical regulations
\n\* “Derive” a work item: A team that is responsible for a derivation step in a trace for work items of a specific type looks for all unresolved work items and derives the resolving work items that are part of the own team scope. E.g., an ARC team sees unresolved actions in an operational process an derives all needed system requirements and functions that need to be worked out.
\n\* “Propose” a work item: A team or person receives (assignment) a work item with the status “proposed” from another team, that wants to resolve a dependency. E.g., an OD team designed an action in an operational process and “proposes” (“push”) a system requirement to an ARC team.
\n\* “Design / engineer / work out” a work item: The content of the work item is worked out. (e.g. an operational process with all actions)
\n\* “Break down” (“dependencies defined/delegated”) a work item: All necessary next work items are derived (as proposals) and linked.
\n\* “Resolve” a work item: All necessary next/downwards work items are derived, linked, designed, approved and resolved. E.g., all system functions to fulfil (resolve) an operational requirement are completed.
\n\* “Assign” a work item: The work item is assigned to a team or person that shall design it.
\n\* “Uplink” a work item. A “free floating” (not linked top-down, completely unlinked, or currently invalid) work item (e.g. coming from a bottom-up process or external contributor) is linked “upwards” to a “traced/valid” work item as if it had been derived from it before (e.g. a process is linked ex post to a capability). This is only done if this new link is really needed to resolve the traced work item in a correct way. Afterwards, the following “change action” is executed.
\n\* “Change a work item or link in an existing trace”: Because of a change of a work item or link in a trace the trace gets invalid in some parts(has “suspect links”). It must be reassured that the derivation process in all directions of the trace is correct, or otherwise further work items need to be adapted until all work items are resolved again. E.g., a system requirement lead to 5 system functions that are needed. One system function cannot be implemented. Therefore, all 5 derivations need to be reworked and perhaps all functions need to be deleted and the resolving process starts again. Other example for a suspect link: A function was derived from a requirement. Later the requirement is changed, and the link gets automatically suspect because it is not clear if the function is still implementing the requirement.
\n\* “Approve” a work item (see chapter below). An approval step is done in the ALM on work item level by experts based on their knowledge. The group of approvers is assigned depending on the work item type and content. A single approver agrees with an approval that a work item is correctly derived, designed, and broken. Approvals are done per single work items like requirements, concept paragraphs or operational requirements. Approval sessions are typically done for a selected set of work items.
\n\* “Decide a deliverable”. When all work items for a certain work area / release are resolved and approved, a document is generated for a formal decision process (e.g. an interface specification or a rulebook).
\n\* The typical sequence of actions in a work step is
\n \* Pull a work item or check proposed work items
\n \* Design the work item
\n \* Resolve the work item
\n \* Assign resolving work items to a team or person", -- "A workflow step includes some of the following actions \n\n\* “Create a trace”: The work item on the “top/start” of a trace is created based on external decisions. These are typically\n\n \* Steering group decisions about system capabilities, common business objectives, improvement targets or problem descriptions, operational requirements, new safety requirements, etc. (see chapter below).\n\n \* Technical regulations\n\n\* “Derive” a work item: A team that is responsible for a derivation step in a trace for work items of a specific type looks for all unresolved work items and derives the resolving work items that are part of the own team scope. E.g., an ARC team sees unresolved actions in an operational process an derives all needed system requirements and functions that need to be worked out.\n\n\* “Propose” a work item: A team or person receives (assignment) a work item with the status “proposed” from another team, that wants to resolve a dependency. E.g., an OD team designed an action in an operational process and “proposes” (“push”) a system requirement to an ARC team.\n\n\* “Design / engineer / work out” a work item: The content of the work item is worked out. (e.g. an operational process with all actions)\n\n\* “Break down” (“dependencies defined/delegated”) a work item: All necessary next work items are derived (as proposals) and linked.\n\n\* “Resolve” a work item: All necessary next/downwards work items are derived, linked, designed, approved and resolved. E.g., all system functions to fulfil (resolve) an operational requirement are completed.\n\n\* “Assign” a work item: The work item is assigned to a team or person that shall design it.\n\n\* “Uplink” a work item. A “free floating” (not linked top-down, completely unlinked, or currently invalid) work item (e.g. coming from a bottom-up process or external contributor) is linked “upwards” to a “traced/valid” work item as if it had been derived from it before (e.g. a process is linked ex post to a capability). This is only done if this new link is really needed to resolve the traced work item in a correct way. Afterwards, the following “change action” is executed.\n\n\* “Change a work item or link in an existing trace”: Because of a change of a work item or link in a trace the trace gets invalid in some parts(has “suspect links”). It must be reassured that the derivation process in all directions of the trace is correct, or otherwise further work items need to be adapted until all work items are resolved again. E.g., a system requirement lead to 5 system functions that are needed. One system function cannot be implemented. Therefore, all 5 derivations need to be reworked and perhaps all functions need to be deleted and the resolving process starts again. Other example for a suspect link: A function was derived from a requirement. Later the requirement is changed, and the link gets automatically suspect because it is not clear if the function is still implementing the requirement.\n\n\* “Approve” a work item (see chapter below). An approval step is done in the ALM on work item level by experts based on their knowledge. The group of approvers is assigned depending on the work item type and content. A single approver agrees with an approval that a work item is correctly derived, designed, and broken. Approvals are done per single work items like requirements, concept paragraphs or operational requirements. Approval sessions are typically done for a selected set of work items.\n\n\* “Decide a deliverable”. When all work items for a certain work area / release are resolved and approved, a document is generated for a formal decision process (e.g. an interface specification or a rulebook).\n\n\* The typical sequence of actions in a work step is\n\n \* Pull a work item or check proposed work items\n\n \* Design the work item\n\n \* Resolve the work item\n\n \* Assign resolving work items to a team or person"

### ontorail:ontolex:LexicalConcept lex\_sp-defs-231113:WORKSTEP--\_WORKITEM--CHECK\_lexConcept modifications from lex\_sp-defs-231012:

 == skos:definition => ++ "\* Is the workitem content correct and without inner contradictions?
\n\* Does the workitem content fit to the wokitem type?
\n\* Are all standard or custom workitem fields filled?
\n\* Does the workitem content implement the demands coming to it by existing links?
\n\* Do all mandatory links exist and are they completely fulfilling their role? Does the content fit to them? E.g. a system requirements can be implemented by a function... but it needs to be completely implemented and not partly.", -- "\* Is the workitem content correct and without inner contradictions?\n\n\* Does the workitem content fit to the wokitem type?\n\n\* Are all standard or custom workitem fields filled?\n\n\* Does the workitem content implement the demands coming to it by existing links?\n\n\* Do all mandatory links exist and are they completely fulfilling their role? Does the content fit to them? E.g. a system requirements can be implemented by a function... but it needs to be completely implemented and not partly."