|  |
| --- |
| SESAR_JU |
| ISRM Foundation Rulebook |
| Document information |
| Project Title | ISRM Foundation Rulebook |
| Project Number | 08.03.10 |
| Project Manager | NORACON |
| Deliverable Name | ISRM Foundation Rulebook |
| Deliverable ID | D45 |
| Edition | 00.08.00 |
| Template Version | 03.00.00 |
| Task contributors  |
| DFS, EUROCONTROL, NORACON, NATMIG, FINMECCANICA |
|  |
| AbstractThe ISRM Rulebook provides rules in order to facilitate the development and maintenance of the ISRM. The rules are intended to be used for modelling, and verification, conformance, and quality check purposes. |

Authoring & Approval

|  |
| --- |
| Prepared By - Authors of the document. |
| Name & Company | Position & Title | Date |
| Bjørn Solberg / NORACON | P08.03.10 member | 29/04/2016 |
| Oliver Schrempf / DFS | P08.03.10 member | 17/05/2016 |

|  |
| --- |
| Reviewed By - Reviewers internal to the project. |
| Name & Company | Position & Title | Date |
| Walter Van Hamme / EUROCONTROL | 8.3.10 Contributor | 11/05/2016 |
| Serena Rubbioli / IDS | 8.3.10 Contributor | 23/05/2016 |
| Tom Erik White / NORACON | 8.3.10 Contributor | 23/05/2016 |

|  |
| --- |
| Reviewed By - Other SESAR projects, Airspace Users, staff association, military, Industrial Support, other organisations. |
| Name & Company | Position & Title | Date |
|  |  |  |

|  |
| --- |
| Approved for submission to the SJU By - Representatives of the company involved in the project. |
| Name & Company | Position & Title | Date |
| Tord Pola/NORACON | Project Manager | 31/05/2016 |
|  |  |  |

|  |
| --- |
| Rejected By - Representatives of the company involved in the project. |
| Name & Company | Position & Title | Date |
|  |  |  |

|  |
| --- |
| Rational for rejection |
| None. |

Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Edition | Date | Status | Author | Justification |
| 00.01.04 | 09/09/2011 |  |  | Reworked according to new modelling guidelines |
| 00.01.06 | 12/3/2013 |  | Bjørn Solberg | Excluded instances, boundaries, texts and notes from the element naming conventions |
| 00.02.00 | 07/06/2013 |  | Bjørn Solberg | Edits to harmonize with other foundation documents and for ISRM 1.0 |
| 00.03.00 | 21/01/2014 |  | Are Kjæraas | Edits to harmonize with other foundation documents and for ISRM 1.1 |
| 00.03.01 | 18/03/2014 | Draft | Are Kjæraas | Implementation of Change Requests CR-3  |
| 00.03.10 | 09/04/2014 | Final | Oliver Schrempf | Preparation for Release |
| 00.04.00 | 16/06/2014 | Draft | Are Kjæraas / Bjørn Solberg | Implementations of Change Requests for ISRM 1.2 |
| 00.04.01 | 26/08/2014 | Final | Are Kjæraas / Bjørn Solberg / Oliver Schrempf | Implementations according to the SJU Review, SMT3 decisions, CR\_26, CR\_58 |
| 00.05.00 | 16.12.2014 | Final | Oliver SchrempfBjørn SolbergSvein G. Johnsen | CR-5 reorganize IER placementCR-43 Make Capability mapping mandatoryCR-49, CR-69, CR-75, CR-76, CR-83, CR-84, CR-85, CR-86, CR-89, |
| 00.06.00 | 10.06.2015 | Final | Oliver SchrempfSvein G. JohnsenTom Erik WhiteGianluca MarrazzoOliver KruegerBjørn SolbergTord Pola | CR-96 Fix requested linkage type for EATMA ElementsCR-16 Alignment of rule namesCR-102 Align AIRM tracing documentationCR-104 Rule 575 set to semi-automaticAdditional rule updates |
| 00.06.01 | 29.09.2015 | Hot Fix | Oliver Schrempf | Fix for ambiguous rule texts SM 290 and SM 571 |
| 00.07.00 | 15.12.2015 | Final | Oliver SchrempfGianluca MarrazzoTord PolaTom Erik WhiteBjørn SolbergAre KjæraasSvein G. Johnsen | CR-98 Remove Taxonomy from ISRMCR-119 Replace attribute-based NFRs by traces to text-based requirements (SPRs).CR-113 Modelling of Enumeration, plus major improvement and simplification on all payload modelling rules.CR-121 Remove hyperlinks from referencesCR-108 Clarify reason for CLDM out of scope in SM302CR-114 Remove 1st sentence from SM301CR-117 Consistent use of diagram nameCR-123 Describe MEP UsageCR-125 New Rule AC020 to ensure Property Note in every diagram in the Views package of all the servicesCR-130 Cleanup sub-package structureCR-131 Minor update of NC020 rule text  |
| 00.08.00 | 25.05.2016 | Final | Bjørn SolbergOliver Schrempf | Modifications according to decisions during the ISRM Foundation 00.08.00 Kickoff |

Intellectual Property Rights (foreground)

This deliverable consists of SJU foreground.

Table of Contents

[Executive summary 6](#_Toc451951324)

[1 Introduction 7](#_Toc451951325)

[1.1 Purpose of the document 7](#_Toc451951326)

[1.2 Intended readership 7](#_Toc451951327)

[1.3 Inputs from other projects 7](#_Toc451951328)

[1.4 Glossary of terms 7](#_Toc451951329)

[1.5 Acronyms and Terminology 7](#_Toc451951330)

[2 Rules 8](#_Toc451951331)

[2.1 General Rules (GRnnn) 8](#_Toc451951332)

[2.1.1 Packages and Structure 8](#_Toc451951333)

[2.2 Naming conventions (NCnnn) 9](#_Toc451951334)

[2.2.1 Package naming conventions 9](#_Toc451951335)

[2.2.2 General model element naming conventions 9](#_Toc451951336)

[2.2.3 Service and operation naming conventions 9](#_Toc451951337)

[2.2.4 Artefact naming conventions 10](#_Toc451951338)

[2.2.5 Diagram naming conventions 10](#_Toc451951339)

[2.3 Service modelling rules (Rule SMxxx) 10](#_Toc451951340)

[2.3.1 Service Rules 11](#_Toc451951341)

[2.3.2 Interface Rules 11](#_Toc451951342)

[2.3.3 Payload Rules 12](#_Toc451951343)

[2.3.4 Diagram Rules 14](#_Toc451951344)

[3 References 15](#_Toc451951345)

[Appendix A ISRM Diagrams and Elements 16](#_Toc451951346)

List of tables

[Table 1: ISRM Diagrams 16](#_Toc451951347)

[Table 2: ISRM Elements 18](#_Toc451951348)

[Table 3: ISRM Associations 18](#_Toc451951349)

[Table 4: ISRM to SDCM Element Mapping 0](#_Toc451951350)

List of figures

Executive summary

The ISRM Rulebook provides rules and information service (IS) modelling standards to be applied in the modelling activities, in order to facilitate the development and maintenance of the ISRM. The rules are intended to be used for modelling, verification, conformance, and quality check purposes.

# Introduction

The ISRM Rulebook provides rules and information service modelling standards to be applied in the modelling activities, in order to facilitate the development and maintenance of Logical Service Models. The rules are intended to be used for modelling, consolidation, validation and verification, conformance, and quality check purposes.

The use and conformance to the standards will ensure consistent and high quality Logical Service Models and will support effective Validation &Verification, conformance and quality check processes.

## Purpose of the document

This document compiles rules applicable for the ISRM. While the ISRM Modelling Guideline [1] covers the process of how to create and edit content in ISRM this document include rules for the produced content. These rules are described in a format making it feasible to verify, manually or automatically, relevant parts of ISRM against them.

The rules are organised in 3 different sections:

* General Rules
* Naming Conventions
* Service Modelling Rules

## Intended readership

The ISRM Foundation Rulebook target audiences include:

* Service Architects
* Service Reviewers
* Information Architects

## Inputs from other projects

N/A

## Glossary of terms

N/A

## Acronyms and Terminology

| Term | Definition |
| --- | --- |
| AIRM | ATM Information Reference Model |
| ATM | Air Traffic Management |
| EATMA | European ATM Architecture |
| ICAO | International Civil Aviation Organization |
| ISRM | Information Services Reference Model |
| SESAR | Single European Sky ATM Research Programme |
| SJU | SESAR Joint Undertaking (Agency of the European Commission) |

# Rules

The rules define various aspects of how the modelling work shall be executed in order to facilitate the development and maintenance of the ISRM. A rule is automatically, semi-automatically or manually verifiable.

Automatic verification is achieved by execution of verification scripts[[1]](#footnote-2) that are available the Sparx Enterprise Architect tool. The rules that have corresponding scripts to automatically verify them are denoted by “**[A]**” in their titles below.

Semi-automatic verification is performed by a combination of execution of verification scripts and manual verification. The rules verifiable semi-automatically is denoted by "**[S]**" in their titles below.

For manually verifiable rules, "**[M]**" is denoted in their titles.

## General Rules (GRnnn)

Rule GR010: Modelling Language [M]

All ISRM services shall be modelled using the Unified Modelling Language (UML).

Rule GR020: Use of language in names and text [M]

Names of and text for (e.g. descriptions in Notes) model elements shall be in the English language following preferred terms as identified by Rules. Where conflicting spellings exist, the spelling listed as the primary British spelling shall be used.

### Packages and Structure

Rule GR100: Packaging of model elements [M]

All ISRM model elements shall be located inside a UML::Package in the model.

Rule GR110: Service root package [M]

All ISRM model elements including packages shall be located inside one single UML::Package which is referred to as root package.

Rule GR120: Model package properties [A]

The service root package shall have the following properties:

* Version: X.0 for an approved[[2]](#footnote-3) (from the package status) version, X.Y for a non-approved (proposed) version. X and Y are integers >= 0.
* Author: Author shall give the name of the person and organization responsible for creating this version.

Rule GR130: Service root package structure [A]

The Service root package shall include the following sub packages:

* Elements
* Diagrams

Rule GR140: Elements package sub-structure [A]

The sub-package ‘Elements’ shall include following sub packages:

* Service
* Payload
* Event Trace
* Requirements (optional for when Requirements are present in the model)
* Abbreviations (optional for when Abbreviations are used in the model)

## Naming conventions (NCnnn)

This section contains rules related to naming conventions for different categories of model elements and artefact.

### Package naming conventions

Rule NC100: Service package name [A]

The root package shall be named by the name of the service followed by the suffix 'Service'.

### General model element naming conventions

This section refers standards related to naming of model elements (packages, elements, diagrams) in the ISRM. The set of elements excluded from this is instances, boundaries, text and notes, along with well-known packages in the model structure (Elements, and Diagrams).

Rule NC200: Verbs used in names [M]

Verbs used in names shall be in the present tense.

Rule NC210: Allowed characters in names [A]

All Names of elements other than Requirements, Abbreviations, or Event Trace elements shall not contain special characters. Allowed characters are [a-z], [A-Z], [0-9],”-“,”\_”

Rule NC220: Use of UpperCamelCase in names [S]

All words in names (except attribute, property, operation, parameter and requirement names) must be in UpperCamelCase. Acronyms must be kept in their original form (upper case).

Examples:

* AsInThisSentence
* ATMNetworkManagement

Rule NC230: Use of lowerCamelCase in names [S]

All words in attribute, property, operation and parameter names must be in lowerCamelCase (except for Enumerations). For allowed acronyms (see rule NC100), they must be kept in their original form, except one that is in the beginning of the word – it must be all lowercase.

Examples:

* asInThisSentence
* amanStatusValue
* positionInILS.

Rule NC240: Use of abbreviations in names [S]

Abbreviations and acronyms shall not be used in names of model elements except where they appear in the AIRM List of Abbreviations or as model elements in an Abbreviation package as described in Table 2: ISRM Elements.

### Service and operation naming conventions

Rule NC300: Uniqueness of service names [A]

All service names shall be unique within the ISRM; no two services shall have the same name.

Rule NC310: Service naming [M]

A service name shall contain the name of the Concept it encapsulates, always include a Noun, and not end with the suffix 'service'

Rule NC320: Service operation naming [S]

A service operation name shall consist of a verb+[optional Adjective]+Noun.

Examples:

* getAlerts
* requestTrajectoryAnalysis
* publishAirportMETInducedCapacity
* setCoordinationAndTransferData
* proposeARESDeActivation

### Artefact naming conventions

Note: The types of the artefacts to be delivered as part of the service design is not finally decided. It will be influenced by the service framework and underlying technology choices yet to be made by other SJU projects. The below rules for naming of artefacts are to be considered if the artefact is delivered.

Rule NC400: Name of XMI files [M]

The XMI file for an exported package shall be named according to the name of the exported package, possibly suffixed by version or status information.

### Diagram naming conventions

Rule NC500: Naming diagrams for a service [A]

For each service, each diagram present shall have a name starting with the name of the service, or some consistent (across the diagrams) short form of the service name. Then follows the name of the diagram. Finally, a free text supplement may be added to differentiate between several similar service diagrams.

Allowed diagram naming in the “Diagrams” package:

* <ServiceName> Requirements Traceability
* <ServiceName> Interface Parameter Definition
* <ServiceName> Interface Definition
* <ServiceName> Event Trace Description

Service naming examples:

* SharedFlightObject Interface Definition
* ATCFlightObjectControl Event Trace Description for i4D

## Service modelling rules (Rule SMxxx)

For rules pertaining to model elements, the set of model elements excluded from this is instances, boundaries, text and notes.

Rule SM010: Service abstraction [M]

The service model shall not contain model constructs with a purpose to support specific data structures, algorithms, systems or implementation technologies. Modelling is done at a logical level.

*Note*: Adding data or technology specific model constructs to a model in general imposes constraints that may make a model unnecessary dependent on implementation decisions. The ISRM shall be focused on describing strategic and operational needs and interactions independent of implementation type, language and technological platforms.

Rule SM020: Model element definition [A]

Model elements as indicated in Table 2 shall have a definition. The definition of model elements shall be stored in the Notes field of that element. (see Table 2: ISRM Elements column *Def. needed*)

Rule SM030: Model element placement [A]

All model elements given in Table 2 shall be placed in the sub packages indicated by the table. (See Table 2: ISRM Elements column *UML Package*)

Rule SM040: Allowed Stereotypes and UML types per package [A]

Only the stereotypes and UML types listed in Table 2 are allowed to be used for elements in the sub packages. (See Table 2: ISRM Elements columns *UML Type*, *Stereotype,* and *UML package*)

Rule SM050: Elements visibility [A]

All elements that are defined within the service model must appear on the diagrams given in Table 2 in the given multiplicity. (See Table 2: ISRM Elements columns *On Diagrams*)

### Service Rules

Rule SM100: Service to requirements mapping [A]

The Service Element <<Service>> may be linked by a dependency relation to the information exchange and other requirement(s) the service exposes, i.e. a dependency relation with a <<satisfy>> stereotype. The links, when present, shall be established in a Requirements Traceability diagram.

Rule SM110: Service element stereotype and placement [A]

The Service Model shall have exactly one service element stereotyped <<Service>> located in the Service package in Service Elements Package structure.

Rule SM120: Service port(s) attached to service elements [A]

The Service Element <<Service>> must have at least one port attached to it.

Rule SM130: Service port MEP usage [A]

The Service Element port must be stereotyped to indicate the Message Exchange Pattern (MEP) it is based on. The list of applicable stereotypes is given in Table 2.

### Interface Rules

Rule SM200: Provider Interface [A]

The service port(s) must be linked to a provided interface. The link shall be a UML realization with stereotype <<provide>>. (See Table 3: ISRM Associations)

Rule SM210: Consumer Interface [A]

The service port(s) may (depending on the used MEP) be linked to a required interface. The link shall be a UML dependency with stereotype <<require>>. (See Table 3: ISRM Associations)

Rule SM220: Consumer Interface linkage [A]

The service consumer shall (depending on the used MEP) be linked to interface(s). The link(s) shall be of UML type realization with stereotype <<provide>> or of UML type dependency with stereotype <<require> respectively. (See Table 3: ISRM Associations)

Rule SM225: Operations attached to service interfaces [A]

All Service Interfaces must have one or more operations attached to it.

Rule SM230: Service interface operations payload design [A]

The parameters and return values used by the operations must be typed (classified) by Payloads (<<Message>>) defined in the model within the services package structure.

Rule SM240: Synchronous Request/Reply MEP [A]

A «SyncReqRep» Port shall provide **one Provider Interface** and require **no Consumer Interface.**

The Provider Interface operations shall contain an input parameter and a return value.

Rule SM250: Asynchronous Request Reply MEP [A]

An «AsyncReqRep» Port shall provide **one Provider Interface** and require **one Consumer Interface.**

The Provider Interface operations shall contain an input parameter and **no** return value.

The Consumer Interface shall contain *callback* operations with an input parameter and **no** return value.

Rule SM260: Publish/Subscribe Push MEP [A]

A «PubSubPush» Port shall provide **one Provider Interface** and require **one Consumer Interface.**

The Provider Interface name shall end with the term “Publisher” and shall contain subscription and one unsubscription operations with an input parameter and optionally return values.

Allowed verbs for the Provider Interface Operations are “subscribe” and “unsubscribe” respectively.

The Consumer Interface name shall end with the Term “Subscriber” and shall contain publication *callback* operations with an input parameter and **no** return value.

The allowed verb for the publication operation is “publish”.

Rule SM270: Publish/Subscribe Pull MEP [A]

A «PubSubPull» Port shall provide **one Provider Interface** and require **no Consumer Interface.**

The Provider Interface name shall end with the term “Publisher” and shall contain one subscription, one unsubscription, and one pull operation with an input parameter and optionally return values.

Allowed verbs for the Provider Interface Operations are “subscribe”, “unsubscribe”, and “pull” respectively.

Rule SM280: One-Way MEP [A]

A «OneWay» Port shall provide **one Provider Interface** and require **no Consumer Interface.**

The Provider name shall end with the term “Listener” and Interface operations shall contain an input parameter and **no** return value.

The allowed verb for the One-Way operation is “post”.

### Payload Rules

Rule SM300: Service payload representation [A]

The payload (as parameters or return value) shall be modelled as a UML tree structure built on the following elements:

* + - Payload elements (UML classes) with stereotype <<Message>> or <<DataEntity>> where a <<Message>> is always the root of the tree structure which is exposed as type of a parameter or return value;
		- Enumerations (UML type)

Each element with stereotype <<Message>> or <<DataEntity>> may contain attributes, unless representing a part of a standard exchange model.

Rule SM310: Payload specialization [A]

A Payload may be a specialization of another Payload. The Payload being specialized must reside within the same service package structure as the specializing Payload. The specialization can only be between Payloads of the same stereotype (i.e. <<Message>> or <<DataEntity>>).

Rule SM320: Payloads structure [A]

A Payload entity may have a composit Aggregation Association (also known as composition) with one or more Payloads. In such cases the relationship shall be represented with the name given in the Target Role of the composition connector.

Rule SM330: Message structure [A]

A <<Message>> Payload may connect to <<DataEntity>> Payloads through compositions. A <<Message>> Payload shall not connect to another <<Message>> Payload through composition.

Rule SM340: DataEntity structure [A]

A <<DataEntity>> Payload may connect to other <<DataEntity>> Payloads using compositions. <<DataEntity>> Payloads shall not connect to <<Message>> Payloads through compositions.

Rule SM350: Payload representing parts of standard exchange models [S]

A Payload containing no attributes and having no outbound composition-connectors or specialization-connectors is considered a part of an existing exchange model. It must contain an IMDefinitionTrace tagged value referencing its counterpart entity or property inside the AIRM Information Model (IM). This tagged value shall be set to a valid URN within the AIRM IM.

The Payload shall contain a Notes description explaining what is the underlying data exchange model (name and edition) and what is the class, within the exchange model, representing the service payload. See detailed requirements in the AIRM Compliance Handbook.

Rule SM360: Payload attribute semantic trace to AIRM CLDM [A]

Every Payload attribute must have a *CLDMSemanticTrace* tagged value set with a valid URN to the appropriate AIRM CLDM attribute or role, indicating it has a counterpart in the CLDM. If the data item in question is not in scope for AIRM, then the *CLDMSemanticTrace* tagged value must be set to “CLDM\_out\_of\_scope”. See detailed requirements in the AIRM Compliance Handbook

E.g. “urn:x-ses:sesarju:airm:v400:ConsolidatedLogicalDataModel:SubjectFields:BaseInfrastructure: AerodromeInfrastructure:RunwayDirection@designator”

Rule SM370: Typing of Payload attribute [A]

Every Payload attribute shall be left without a type with the following exceptions:

* If the Payload attribute has its *CLDMSemanticTrace* set to “CLDM\_out\_of\_scope” then it must be typed by a base or foundation type available in the AIRM.
* If the Payload attribute has a value constrained by an AIRM- or ISRM-specific Enumeration, then it must be typed by the corresponding Enumeration.

Rule SM380: Payload attribute definition trace to AIRM IM [A]

Every Payload attribute may have an optional *IMDefinitionTrace* tagged value referencing its counterpart entity or property inside the AIRM Information Model (IM). If present, this tagged value shall be set to a valid URN within the AIRM IM.

Rule SM390: Payload attribute context trace to AIRM CLDM [A]

Every Payload attribute may have one or more optional *CLDMContextTrace* tagged values, each referencing its counterpart entity or property inside the AIRM CLDM. If present, each tagged value shall set to a valid URN within the AIRM CLDM.

### Diagram Rules

Rule SM400: Diagram existence [A]

The diagrams defined in Table 1 must exist in the multiplicity specified in the table. (See Table 1: ISRM Diagrams column *Occur*.)

Rule SM410: Diagram elements [A]

All diagrams shall contain the elements specified in Table 2 for the specific type of Diagram. (See Table 2: ISRM Elements column *On Diagrams*)

Rule SM420: Placement of diagrams [A]

All service diagrams must be placed within the Diagrams package of the service package structure.

Rule SM430: Meta information in every diagram [A]

All diagrams must include visible meta information at least showing the name of the diagram, author, when the diagram was created and when it was last updated. This may be present in a Note, Diagram info, or as a Property Note on the diagram.

# References

1. 08.03.10, ISRM Modelling Guidelines, D45, 31/05/2016
2. ISRM Diagrams and Elements

This Appendix contains tables listing elements to be used in a service model. The tables are reference by the rules given in this document.

The following notation is used to express the multiplicity of occurrence of model elements and diagrams:

|  |  |
| --- | --- |
| 0 | an element of this type never appears on a diagram of this type |
| 0+ | an element of this type is optional on a diagram of this type |
| all | the (all) element(s) of this type appear on all diagram of this type |
| 1+ / .. | any element of this type must appear at least once on a diagram of this type |
| .. / 1+ | a diagram of this type must at least show one element of this type |
| .. / 0+ | a diagram of this type may show elements of this type |

The following Abbreviations are used:

UCC = UpperCamelCase

LCC = lowerCamelCase

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Name** | **Package** | **Occur.** |
| Class Diagram | <ServiceName> Requirements Traceability <free suffix> | Diagrams | 0..\* |
| Class Diagram | <ServiceName> Interface Definition <free suffix> | Diagrams | 1..\* |
| Class Diagram | <ServiceName> Interface Parameter Definition <free suffix> | Diagrams | 1..\* |
| Sequence Diagram | <ServiceName> Event Trace Description <free suffix> | Diagrams | 1..\* |

Table 1: ISRM Diagrams

| **Element Name** | **UML Type** | **Stereotype** | **UML package** | **Tag values** | **naming** | **Def.needed** | **Comment** | **On Diagrams** (see Table 1: ISRM Diagrams) |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Service | Class | «Service» | Service |  | UCC | yes |  | all | all | 0 | all |
| Port | -port | «SyncReqRep»«AsyncReqRep»«PubSubPush»«PubSubPull»«OneWay» | Service |  | UCC | yes | On Service  | 0 | 1+ / 1+ | 0 | 1+ / 1+ |
| Interface | Interface | - | Service |  | UCC |  yes |  | 0 | 1+ / 1+ | 0 | 0 |
| Operation | -operation | - | Service |  | LCC | yes | On Interface; return type with Message | 0 | 1+ / 0+ | 0 | 1+ / 1+ |
| Parameter | -parameter | - | Service |  | LCC |  | On Operation; typed with Message | 0 | 0+ | 0 | 0+ |
| Consumer | Actor | «ServiceConsumer» | Service |  | UCC + "consumer" | yes |  | 0 | 1+ / 1+ | 0 | 1+ / 1+ |
| Message | Class | «Message» | Payload | AIRMtraces… | UCC | yes | must be used as parameter type or return type in at least one operation | 0 | 1+ / 0+ | 1+ / 0+ | 1+ |
| DataEntity | Class | «DataEntity» | Payload | AIRMtraces… | UCC | yes | Any other payload element; cannot be used as operation parameter type or return type  | 0 | 0 | 1+ / 0+ | 0 |
| Attribute | -attribute | - | Payload | AIRMtraces… | LCC | yes | On Message and DataEntity | 0 | 0 | 1+ / 0+ | 0 |
| Enumeration | Enumeration | - | Payload | AIRMtraces… | UCC | yes |  | 0 | 0 | 1+ / 0+ | 0 |
| Requirement | Class | «Requirement» | Requirements | refLabel, refSource, refURL, reqType | free |  | IER or NFR requirement | 1+ / 1+ | 0 | 0 | 0 |
| Abbreviation | Class | «Abbreviation» | Abbreviations |  | free | yes | Contains an abbreviation as name and its definitions and source as note | 0 | 0 | 0 | 0 |
|  | Fragment |   | Event Trace |  | free |  | Element for sequence diagram | 0 | 0 | 0 | 1+ / 0+ |
|  | Lifeline |  | Event Trace |  | free |  | Element for sequence diagram | 0 | 0 | 0 | 1+ / 0+ |
|  | Boundary |  | Event Trace |  | free |  | Element for sequence diagram | 0 | 0 | 0 | 1+ / 0+ |
|  | Control |  | Event Trace |  | free |  | Element for sequence diagram | 0 | 0 | 0 | 1+ / 0+ |

Table 2: ISRM Elements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UML Type** | **Stereotype** | **From** | **To** | **Comment** | **On Diagrams** |
| Composition | - | Message / DataEntity | DataEntity |  | 0 | 0 | 1+ / 0+ | 0 |
| Generalization | - | Message | Message |  | 0 | 0 | 1+ / 0+ | 0 |
| Generalization | - | DataEntity | DataEntity |  | 0 | 0 | 1+ / 0+ | 0 |
| Realization | «provide» | Port / Consumer | Interface |  | 0 | 1+ / 0+ | 0 | 0 |
| Dependency | «require» | Port / Consumer | Interface |  | 0 | 1+ / 0+ | 0 | 0 |
| Dependency | «satisfy» | Service | Requirement |  | 1+ / 1+ | 0 | 0 | 0 |

Table 3: ISRM Associations

| **ISRM Element** | **SDCM Element** | **European Extension** |
| --- | --- | --- |
| Service | Service  |  |
| Port |   | Name: **Port**Description: A port is a grouping of interfaces with an associated MEP.  |
| Interface | Interface |  |
| Operation | Operation |  |
| Parameter | Message |  |
| Consumer | Consumer |  |
| Message | Payload |   |
| DataEntity | Data Entity |  |
| Attribute |  | Name: **Payload Attribute**Description: A Payload Attribute specifies the content of Payload and DataEntities |
| Enumeration |  | Name: **Payload Enumeration**Description: A Payload Enumeration is a defined List of applicable values for Payload Attributes |
| Requirement | N/A | N/A |

Table 4: ISRM to SDCM Element Mapping

-END OF DOCUMENT-

1. Note that no verification scripts are available yet for the rules defined in this version (00.08.00) for the rulebook. The primer contains an archive of the scripts used for version 00.07.00. [↑](#footnote-ref-2)
2. The status „approved“ has to be identified by the authority which manages the service lifecycle. [↑](#footnote-ref-3)