ETSI TS 103 548 V1.2.1 (2024-01)

SmartM2M;
SAREF reference ontology patterns

Reference
RTS/SmartM2M-103548v121

Keywords
data, interoperability, IoT, oneM2M, ontology,
SAREF, semantic

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Machine-to-Machine communications (SmartM2M).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies the set of SAREF reference ontology patterns [2] for the modelling and the description of any kind of applications-related data/information/systems. The SAREF reference ontology patterns can be applied to different verticals to provide the SAREF suite of ontologies [1] a homogeneous and predictable structure, and to achieve higher semantic interoperability. They extend and are aligned with the core of the following ontologies: W3C&OGC SOSA (Sensing, Observation, Sampling and Actuation) and SSN (Semantic Sensor Network) [i.1], QUDT (Quantity, Unit, Dimension and Type) [i.2], DUL (Dolce+DnS Ultralite Ontology) [i.3]. The present document is based on the requirements and guidelines defined in ETSI TR 103 549 [i.4] and ETSI TR 103 781 [i.5].

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 103 264 (V3.2.1): "SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping".

[2] ETSI TS 103 673 (V1.2.1): "SmartM2M; SAREF Development Framework and Workflow, Streamlining the Development of SAREF and its Extensions".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] W3C® Recommendation, 19 October 2017: "Semantic Sensor Network Ontology", A. Haller, K. Janowicz, S. Cox, D. Le Phuoc, K. Taylor, M. Lefrançois, R. Atkinson, R. García-Castro, J. Lieberman, C. Stadler.

[i.2] J. Hodges, R. Hodgson, S. Ray, N. Car, S. Chalk, S. Cox, C. Paul, T. Smith: "QUDT; Quantities, Units, Dimensions and Types". QUDT.org, 10.25504/FAIRsharing.d3pqw7, 2011.

[i.3] C. Masolo, S. Borgo, A. Gangemini, N. Guarino, A. Oltramari, and L. Schneider. "The WonderWeb Library of Fundational Ontologies and the DOLCE ontology". Technical report, LOAISTC, 2003.

[i.4] ETSI TR 103 549: "SmartM2M; Guidelines for consolidating SAREF with new reference ontology patterns, based on the experience from the ITEA SEAS project".

[i.5] ETSI TR 103 781: "SmartM2M; Study for SAREF ontology patterns and usage guidelines".

[i.6] ETSI TS 103 410-1: "SmartM2M; Extension to SAREF; Part 1: Energy Domain".

[i.7] ETSI TS 103 410-2: "SmartM2M; Extension to SAREF; Part 2: Environment Domain".

[i.8] ETSI TS 103 410-3: "SmartM2M; Extension to SAREF; Part 3: Building Domain".

[i.9] ETSI TS 103 410-4: "SmartM2M; Extension to SAREF; Part 4: Smart Cities Domain".

[i.10] ETSI TS 103 410-5: "SmartM2M; Extension to SAREF; Part 5: Industry and Manufacturing Domains".

[i.11] ETSI TS 103 410-6: "SmartM2M; Extension to SAREF; Part 6: Smart Agriculture and Food Chain Domain".

[i.12] ETSI TS 103 410-7: "SmartM2M; Extension to SAREF; Part 7: Automotive Domain".

[i.13] ETSI TS 103 410-8: "SmartM2M; Extension to SAREF; Part 8: eHealth/Ageing-well Domain".

[i.14] ETSI TS 103 410-9: "SmartM2M; Extension to SAREF; Part 9: Wearables Domain".

[i.15] ETSI TS 103 410-10: "SmartM2M; Extension to SAREF; Part 10: Water Domain".

[i.16] ETSI TS 103 410-11: "SmartM2M; Extension to SAREF; Part 11: Lift Domain".

[i.17] ETSI TS 103 410-12: "SmartM2M; Extension to SAREF; Part 12: Smart Grid Domain".

[i.18] ETSI TR 103 411 (V1.1.1): "SmartM2M; Smart Appliances; SAREF extension investigation".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

ontology: formal specification of a conceptualization, used to explicitly capture the semantics of a certain reality

ontology design pattern: reusable solutions intended to simplify ontology development and support the use of semantic technologies by ontology engineers

ontology pattern: combination of an ontological definition and specification of how to apply it to different domains

smart application: application using devices which have the ability to communicate with each other and which can be controlled

3.2 Symbols

For the purposes of the present document, the following symbols apply:

RN	Wire 'R' (phase R) to wire 'N' (Neutral)
SN	Wire 'S' (phase S) to wire 'N' (Neutral)
TN	Wire 'T' (phase T) to wire 'N' (Neutral)

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

DL	Description Logics
DP	Datatype Property
DUL	DOLCE+DnS Ultralite ontology
HTPMP	Heat Pump
IoT	Internet of Things

IRI	Internationalized Resource Identifier
OGC	Open Geospatial Consortium
OP	Object Property
OWL	Web Ontology Language
OWL-DL	Web Ontology Language - Description Logics
QUDT	Quantities, Units and DataTypes
RDF	Resource Description Framework
SAREF	Smart Applications REFerence ontology
SOSA	Sensor, Observation, Sample and Actuator
SSN	Semantic Sensor Networks
TR	Technical Report
TS	Technical Specification
USB	Universal Serial Bus
W3C®	World Wide Web Consortium

4 Introduction

4.1 Definition and Purpose of the SAREF Reference Ontology Patterns

SAREF V3.2.1 [1] is a reference ontology for the IoT developed by ETSI SmartM2M in close interaction with the industry. SAREF contains core concepts that are common to several IoT domains and, to be able to handle specific data elements for a certain domain, dedicated extensions of SAREF have been created, namely SAREF4ENER [i.6], SAREF4ENVI [i.7], SAREF4BLDG [i.8], and SAREF4CITY [i.9], SAREF4INMA [i.10], SAREF4AGRI [i.11], SAREF4AUTO [i.12], SAREF4EHAW [i.13], SAREF4WEAR [i.14], SAREF4WATR [i.15], SAREF4LIFT [i.16], SAREF4GRID [i.17]. Each domain can have one or more extensions, depending on the complexity of the domain. As a reference ontology, SAREF serves as the means to connect the extensions in different domains. The earlier document ETSI TR 103 411 [i.18] specifies the rationale and methodology used to create, publish and maintain the SAREF extensions.

The SAREF reference ontology patterns consist of a combination of an ontological definition and a specification of how to apply it in SAREF extensions and applications [2]. Their aim is to help ensuring a homogeneous structure of the overall SAREF ontology, speed up the development of extensions, and improve semantic interoperability. The set of SAREF ontology patterns is summarized at https://saref.etsi.org/patterns/. Each self-contained pattern is described by its unique name, a functional description, a graphical diagram, and the RDF source of the pattern. Browse the entire list to make effective use of the patterns during ontology development.

The present document specifies 14 SAREF reference ontology patterns that can be applied to different domains. Some of these patterns are related to the SAREF core ontology, and are related to the application of classes and properties that frequently appear when developing IoT applications. In other cases, a set of patterns has been defined to provide a general design solution for some cross-domain aspect, such as in SAREF4SYST; in this case a dedicated ontology has been developed to support the application of such patterns:

- SAREF core reference ontology patterns (specified in clause 5)
- SAREF pattern for feature kinds and features of interest
- SAREF pattern for devices
- SAREF pattern for tasks
- SAREF pattern for commodities
- SAREF pattern for properties
- SAREF pattern for states
- SAREF patterns for functions
- SAREF patterns for commands

- SAREF patterns for services and operations
- SAREF patterns for procedure executions
- SAREF patterns for profiles
- SAREF4YST reference ontology patterns (specified in clause 6)
- SAREF4SYST patterns for systems and sub-systems
- SAREF4SYST patterns for connections between systems
- SAREF4SYST patterns for connection points of systems

4.2 Namespaces

For the purposes of the present document, the namespaces from Table 1 of ETSI TS 103 264 [1], apply, along with those listed in Table 1.

Table 1: Prefixes and namespaces used in the present document

Prefix	Namespace
s4syst	https://saref.etsi.org/saref4syst/
ex	https://example.org/ *<solely for the purpose of examples>*

5 SAREF Core Patterns

5.1 Introduction

The following patterns are related to the SAREF core ontology and deal with the application of classes and properties from that ontology that frequently appear when developing IoT applications.

The relevant namespaces and ontological definition are those of the SAREF core ontology [1] and defined in Table 1. These patterns should be used in every SAREF extension.

In general, the identifiers (i.e. IRI) of the new entities (classes, properties, and individuals) shall be defined following the Term IRI requirements (see ETSI TS 103 673 [2]), and shall follow the term documentation requirements (i.e. have at least a label and a comment; see ETSI TS 103 673 [2]).

SAREF extensions and application should not adopt modelling choices that would otherwise conflict with those listed in the clauses 5.2 to 5.12.

5.2 SAREF pattern for features

5.2.1 Introduction

Class `saref:FeatureOfInterest` represents any real world entity from which a property or a state may be acted upon, such as observed and controlled. An instance of `saref:FeatureOfInterest` represents one specific real world entity.

`Class saref:FeatureKind` allows to describe kinds of features of interest, with common properties having the same value, and common states being the same. An instance of `saref:FeatureKind` represents an archetype of real world entities, for example to populate product catalogs.

Figure 1 illustrates the main classes and properties in the SAREF Core patterns for Feature kinds and features of interest.

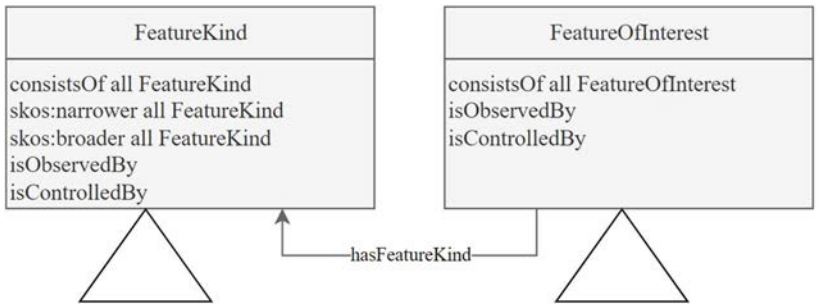

Figure 1: SAREF Core pattern for Features: feature kinds and features of interest

5.2.2 Feature kinds

5.2.2.1 Creating feature kinds

SAREF extensions should create instances of the class `saref:FeatureKind`, such that they can be reused in different SAREF applications.

SAREF applications may also create specific instances of the class `saref:FeatureKind`.

The identifier and the label of an instance of feature kind should not contain "feature". Its comment should mention "kind". The identifier should not end with "Kind".

EXAMPLE:
```
ex:1000x2000mmWindow a saref:FeatureKind ;
    rdfs:label "1000x2000mm window kind"@en ;
    rdfs:comment "The kind of windows with dimensions
 1000x2000mm."@en;
    saref:hasPropertyValue
      [ saref:hasValue 1000 ;
        saref:hasUnit <http://qudt.org/vocab/unit/MilliM> ;
        saref:isValueOfProperty saref:Width ] ,
      [ saref:hasValue 2000 ;
        saref:hasUnit <http://qudt.org/vocab/unit/MilliM> ;
        saref:isValueOfProperty saref:Height ].
```

5.2.2.2 Specializing feature kinds

SAREF extensions may define a taxonomy of feature kinds using `skos:broader and skos:narrower`.

The identifier, the label, and the comment of an instance of `saref:FeatureKind` should explicit its position in the taxonomy of feature kinds.

EXAMPLE:
```
ex:1000x2000mmSmartWindow is a specialization of
ex:1000x2000mmWindow.
ex:1000x2000mmSmartWindow a saref:FeatureKind ;
  rdfs:label "1000x2000mm smart window"@en ;
  skos:broader ex:1000x2000mmWindow ;
  rdfs:comment "The kind of smart windows with dimensions
1000x2000mm, controlled by an electric window actuator."@en ;
  saref:hasPropertyValue
    [ saref:hasValue 1000 ;
      saref:hasUnit <http://qudt.org/vocab/unit/MilliM> ;
      saref:isValueOfProperty saref:Width ] ,
    [ saref:hasValue 2000 ;
      saref:hasUnit <http://qudt.org/vocab/unit/MilliM> ;
      saref:isValueOfProperty saref:Height ] ;
  saref:isControlledBy ex:ElectricWindowActuatorKind .
```

5.2.2.3 Categorizing feature kinds

SAREF extensions and applications may define categories of feature kinds by creating sub-classes of `saref:FeatureKind`.

NOTE: Defining categories of features kinds is especially useful to group those feature kinds that are defined in the same taxonomy.

The identifier, the label and the comment of a sub-class of `saref:FeatureKind` should explicit its position in the hierarchy of feature kind classes. The identifier should end with "Kind".

EXAMPLE: The class `ex:WindowKind` groups all window kinds of a taxonomy `ex:`.

```
ex:WindowKind a owl:Class ;
  rdfs:subClassOf saref:FeatureKind ;
  rdfs:label "Window Kind"@en ;
  rdfs:comment "Category ex:WindowKind groups archetypes of real
world windows."@en .
```

5.2.3 Features of interest

5.2.3.1 Creating features of interest

SAREF extensions should not create specific instances of the class `saref:FeatureOfInterest`, as they are theoretically specific to an application.

NOTE: The only exceptions to this provision are if the feature of interest is intended to be used by most applications of the SAREF extension.

The identifier and the label of an instance of feature of interest should not contain "of interest". The comment may mention "feature of interest".

EXAMPLE: The earth atmosphere is unique and may be used in most applications related to earth satellite observations.

```
  ex:EarthAtmosphere a saref:FeatureOfInterest ;
    rdfs:label "Earth Atmosphere"@en ;
    rdfs:comment "The earth atmosphere."@en .
```

5.2.3.2 Categorizing features of interest

SAREF extensions and applications may define categories of features of interest by creating sub-classes of `saref:FeatureOfInterest`.

NOTE: Defining a category of features of interest is especially useful for multi-typing, or to be able to use it as the domain or range of some relationship to other entities, captured as OPs in the extension.

EXAMPLE 1: The category `ex:Window` is the range of OP `ex:hasExteriorWindow`.

```
ex:Window a owl:Class ;
  rdfs:subClassOf saref:FeatureOfInterest ;
  rdfs:label "Window"@en ;
  rdfs:comment "Class of windows: an opening in the wall or roof
of a building or vehicle, fitted with glass in a frame to admit
light or air and allow people to see out."@en .

ex:hasExteriorWindow a owl:ObjectProperty ;
  rdfs:range ex:Window .
```

It is possible to define hierarchies of feature of interest classes by defining sub-class relationships between them.

It is recommended that the identifier, the label and the comment of a sub-class of `saref:FeatureOfInterest` explicits its position in the hierarchy of feature of interest classes.

SAREF extensions should not place restrictions on the `saref:FeatureOfInterest` class directly.

The main categories a feature of interest belongs to are the classes it explicitly instantiates.

EXAMPLE 2: Feature of interest `<1000x2000mmSmartWindow_az21az4ze1>` is categorized under `ex:Window, and has feature kind ex:1000x2000mmSmartWindow.`

```
<1000x2000mmSmartWindow_az21az4ze1> a ex:Window ;
   saref:hasFeatureKind ex:1000x2000mmSmartWindow ;
   rdfs:label "1000x2000mmSmartWindow #az21az4ze1"@en ;
   rdfs:comment "1000x2000mm smart window with id az21az4ze1"@en ;
```

Further categories are defined transitively through the class hierarchy.

EXAMPLE 3: By being an instance of `ex:Window`, a feature of interest also belongs to the type `saref:FeatureOfInterest`.

5.2.4 Feature kind vs category of features of interest

SAREF extensions can model a category of features of interest either as an instance of `saref:FeatureKind` or as a sub-class of `saref:FeatureOfInterest`. The modelling choice depends on the knowledge representation needs:

- Use an instance of `saref:FeatureKind` to simply attach properties and optionally their values, states, related commodities, or related sensor kinds that can observe or control features of interest having this kind.
- Use a sub-class of `saref:FeatureOfInterest` if the category needs to be defined as the domain or range of some relationship, modelled as an OP in the extension.

In all cases, it shall not happen that the same identifier is used for an instance of `saref:FeatureKind` and a sub-class of `saref:FeatureOfInterest`.

5.2.5 Specializing OPs and DPs related to features

SAREF extensions and applications shall not define sub-properties of the following properties:

- skos:narrower
- skos:broader
- saref:hasManufacturer
- saref:hasModel

There shall be a unique sub-property of `saref:hasFeatureKind`, with identifier `saref:hasDeviceKind.`

5.3 SAREF pattern for devices

5.3.1 Introduction

Class `saref:Device` represents any tangible object designed to accomplish a particular task by performing one or more functions. An instance of `saref:Device` represents one specific real world entity.

The pattern for devices follows the pattern for feature kinds and features of interest from clause 5.2.

Figure 2 illustrates the main classes and properties in the SAREF Core pattern for Devices.

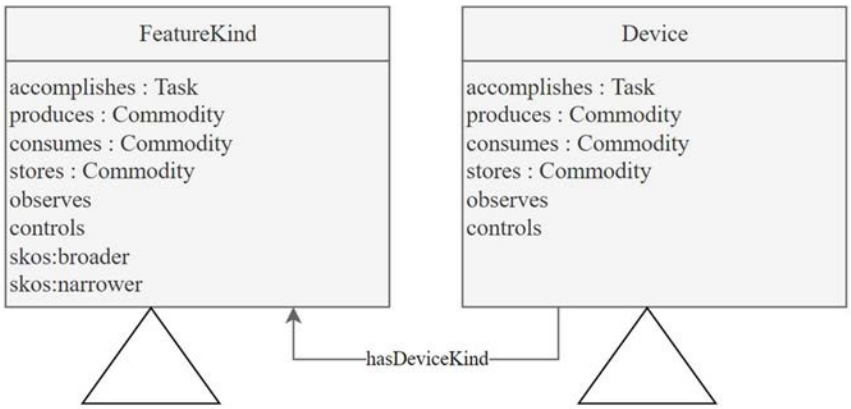

Figure 2: SAREF Core pattern for Devices: device kinds and devices

5.3.2 Creating devices

SAREF extensions should not create specific instances of the class `saref:Device`, as they are theoretically specific to an application.

5.3.3 Categorizing devices

SAREF extensions and applications may define categories of devices by creating sub-classes of `saref:Device`.

NOTE: Defining a category of devices is especially useful for multi-typing, or to be able to use it as the domain or range of some relationship to other entities, captured as OPs in the extension.

EXAMPLE 1:
```
s4watr:WaterDevice categorize devices that are also water assets.

s4watr:WaterDevice rdfs:subClassOf saref:Device ,
s4watr:WaterAsset ;
     rdfs:label "Water device"@en ;
     rdfs:comment "A water device is a device that is also a water
                   asset."@en .
```

It is possible to define hierarchies of device types by defining sub-class relationships between them. SAREF already declares a hierarchy of device types that have to be reused when defining device type hierarchies (e.g. `saref:Sensor or saref:Actuator`).

It is recommended that the identifier, the label and the comment of a sub-class of `saref:Device` explicits its position in the hierarchy of device classes.

EXAMPLE 2:
```
s4watr:WaterMeter rdfs:subClassOf saref:Meter , s4watr:WaterDevice
;
     rdfs:label "Water meter"@en ;
     rdfs:comment "A water meter is an instrument intended to
                   measure continuously, memorize, and display the
                   volume of water passing through the meter."@en .
```

SAREF extensions should not place restrictions on the `saref:Device` class directly.

The main categories a device belongs to are the classes it explicitly instantiates.

EXAMPLE 3: Device `<HTPMP_4A2EF1SI85>` is categorized under `ex:HeatPumpDryer_HTPMP` and `ex:AAHeatPumpDryer`.
```
<HTPMP_4A2EF1SI85> a ex:HeatPumpDryer_HTPMP , ex:AADevice ;
  rdfs:label "HTPMP_4A2EF1SI85"@en ;
  rdfs:comment "Heat pump dryer of kind HTPMP with serial number
4A2EF1SI85"@en ;
```

Further categories are defined transitively through the class hierarchy.

EXAMPLE 4: By being an instance of `s4watr:WaterMeter`, the meter in the example below also belongs to the following device types: `s4watr:WaterDevice`, `saref:Meter` and `saref:Device`.

```
<Meter4837QW123> a s4watr:WaterMeter .
```

5.3.4 Feature kind vs category of device

SAREF extensions can model a category of devices either as an instance of `saref:FeatureKind` or as a sub-class of `saref:Device`. The modelling choice depends on the knowledge representation needs, as defined in clause 5.2.4.

5.3.5 Specializing OPs and DPs related to devices

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:hasDeviceKind
- saref:actsUpon
- saref:isActedUponBy
- saref:observes
- saref:isObservedBy
- saref:controls
- saref:isControlledBy

5.4 SAREF pattern for tasks

5.4.1 Introduction

Tasks are defined as goals for which a device is designed (from a user perspective).

Figure 3 illustrates the main classes and properties in the SAREF Core patterns for Tasks.

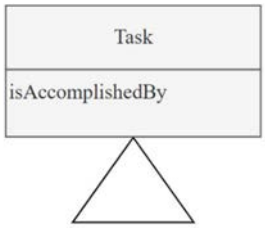

Figure 3: SAREF Core pattern for Tasks

5.4.2 Creating tasks

SAREF extensions should reuse or create specific instances of the class `saref:Task`.

The identifier and the label of an instance of task should start with a capital letter and should not contain "Task". The comment may mention "task".

EXAMPLE:

```
ex:Lighting a saref:Task ;
     rdfs:label "Lighting"@en ;
     rdfs:comment "The task of lighting involves strategically
designing and controlling light sources to achieve optimal
visibility, functionality, and aesthetic appeal in a given
space."@en.
```

5.4.3 Specializing tasks

SAREF extensions may define a taxonomy of tasks using `skos:broader` and `skos:narrower`.

The identifier, the label, and the comment of an instance of `saref:Task` should explicit its position in the taxonomy of tasks.

EXAMPLE: `ex:AmbianceLighting and ex:SecurityLighting` are specializations of `ex:Lighting`.

```
ex:SecurityLighting a saref:Task ;
  skos:broader ex:Lighting ;
  rdfs:label "Security Lighting"@en ;
  rdfs:comment "A kind of task accomplished by lighting devices
intended to deter or detect intrusions or other criminal activity
occurring on a property or site. It can also be used to increase a
feeling of safety."@en .
```

5.4.4 Categorizing tasks

SAREF extensions and applications may define categories of tasks by creating sub-classes of `saref:Task`.

The identifier, the label, and the comment of a sub-class of `saref:Task` should explicit its position in the hierarchy of task classes. The identifier should end with "Task".

EXAMPLE: The class `ex:SafetyTask` groups tasks related to safety.

```
ex:SafetyTask a owl:Class ;
  rdfs:subClassOf saref:Task ;
  rdfs:label "Safety Task"@en ;
  rdfs:comment "Class of tasks related to safety."@en .
```

5.4.5 Specializing OPs and DPs related to tasks

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:accomplishes
- saref:isAccomplishedBy

5.5 SAREF pattern for commodities

5.5.1 Introduction

Class `saref:Commodity` represents marketable items which may be supplied without qualitative differentiation. Commodities may be consumed, produced, or stored, by some feature of interest or device.

Figure 4 illustrates the main classes and properties in the SAREF Core patterns for Commodities.

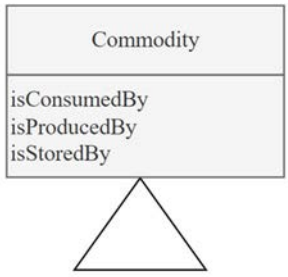

Figure 4: SAREF Core pattern for Commodities

5.5.2 Creating commodities

SAREF extensions should reuse or create specific instances of the class `saref:Commodity`.

The identifier and the label of an instance of commodity should start with a capital letter and should not contain "Commodity". The comment may mention "commodity".

EXAMPLE:
```
ex:Electricity a saref:Commodity ;
     rdfs:label "Electricity"@en ;
     rdfs:comment "A type of energy commodity."@en .
```

5.5.3 Specializing commodities

SAREF extensions may define a taxonomy of commodities using `skos:broader and skos:narrower`.

The identifier, the label, and the comment of an instance of `saref:Commodity` should explicit its position in the taxonomy of commodities.

EXAMPLE: `ex:Propane` is a specialization of `ex:NaturalGas`.
```
ex:Propane a saref:Commodity ;
  skos:broader ex:NaturalGas ;
  rdfs:label "Propane"@en ;
  rdfs:comment "Propane is a type of natural gas commodity."@en .
```

5.5.4 Categorizing commodities

SAREF extensions and applications may define categories of commodities by creating sub-classes of `saref:Commodity`.

The identifier, the label, and the comment of a sub-class of `saref:Commodity` should explicit its position in the hierarchy of commodity classes. The identifier should end with "Commodity".

EXAMPLE: The class `ex:AgriculturalCommodity` groups commodities related to agricultural products.
```
saref:EnergyCommodity a owl:Class ;
  rdfs:subClassOf saref:Commodity ;
  rdfs:label "Energy Commodity"@en ;
  rdfs:comment "Class of energy commodities."@en .
```

5.5.5 Specializing OPs and DPs related to commodities

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:consumes
- saref:isConsumedBy
- saref:produces
- saref:isProducedBy
- saref:stores
- saref:isStoredBy

5.6 SAREF pattern for properties

5.6.1 Introduction

In SAREF, properties refer to the identifiable qualities of features of interest that can be acted upon by devices, such as observed or controlled. While properties can apply to different features of interest, properties of interest are specific to a feature of interest. Property values describe the value for a property.

Figure 5 illustrates the main classes and properties in the SAREF Core patterns for Properties.

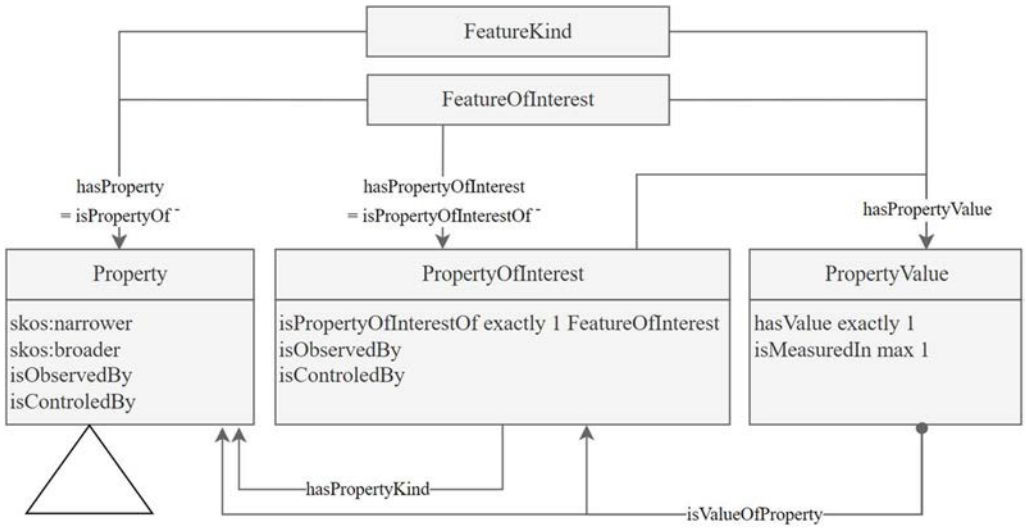

Figure 5: SAREF Core pattern for Properties: properties, properties of interest, property values

5.6.2 Properties

5.6.2.1 Creating properties

SAREF extensions should create instances of the class `saref:Property`, such that they can be reused in different SAREF applications.

SAREF applications may also create specific instances of the class `saref:Property`.

The identifier and the label of an instance of property should not contain "property", and its comment may mention "property kind" or "kind of property".

EXAMPLE:
```
ex:Temperature a saref:Property ;
    rdfs:label "Temperature"@en ;
    rdfs:comment "Temperature is a property kind that expresses
quantitatively the attribute of hotness or coldness."@en .
```

5.6.2.2 Specializing properties

SAREF extensions may define a taxonomy of properties using `skos:broader` and `skos:narrower`.

The identifier, the label, and the comment of an instance of `saref:Property` should explicit its position in the taxonomy of properties.

NOTE: Depending on the use case, the property can explicit the physical dimension its values would have (ex. `ex:AreaProperty`), the feature of interest it applies to (ex. `ex:RoomProperty`), the type of task it relates to (ex. `ex:ComfortProperty`), etc.

EXAMPLE: `ex:WaterTemperature`, is a specialization of `ex:Temperature`.

```
ex:WaterTemperature a saref:Property ;
   skos:broader ex:Temperature ;
   rdfs:label "Water Temperature"@en ;
   rdfs:comment "The temperature of water, a property kind."@en.
```

5.6.2.3 Categorizing properties

SAREF extensions and applications may define categories of properties by creating sub-classes of `saref:Property`.

The identifier, the label, and the comment of a sub-class of `saref:Property` should explicit its position in the hierarchy of property classes.

The identifier, and the label of a property class should end with "Property", and its comment should mention "property class" or "class of properties".

EXAMPLE: The class `ex:WaterFlowProperty` groups properties related to water flow.

```
ex:WaterFlowProperty a owl:Class ;
   rdfs:subClassOf saref:Property ;
   rdfs:label "Water flow property"@en ;
   rdfs:comment "Class of properties related to water flow."@en .
```

5.6.3 Properties of interest

5.6.3.1 Creating properties of interest

SAREF extensions should not create specific instances of the class `saref:PropertyOfInterest`, as they are meant to be specific to an application. The only exceptions to this provision are if the associated feature of interest is intended to be used by most applications of the SAREF extension.

Given a property of interest belongs to exactly one feature of interest, its identifier should consist of the identifier of the feature of interest, followed by character '#' and a fragment identifier. The fragment identifier part of the IRI of a property of interest should not contain "property".

A specific instance of the class `saref:PropertyOfInterest` should link to its kind (an instance of `saref:Property`) using property `saref:hasPropertyKind`, and to the unique feature of interest it is a property of using OP `saref:isPropertyOfInterestOf`.

EXAMPLE:

```
ex:EarthAtmosphere\#Temperature
     a saref:PropertyOfInterest ;
     rdfs:label "Earth Atmosphere Temperature"@en ;
     rdfs:comment "The earth atmosphere temperature, a property of
interest of the earth atmosphere with a unique value in any
spatio-temporal position."en ;
     saref:hasPropertyKind ex:AirTemperature ;
     saref:isPropertyOf ex:EarthAtmosphere .
```

5.6.3.2 Categorizing properties of interest

SAREF extensions should not define sub-classes of `saref:PropertyOfInterest`, as this would duplicate information already part of the taxonomy of properties.

5.6.4 Creating or categorizing property values

SAREF extensions may create specific instances of the class `saref:PropertyValue`, only if they are used as the object of `saref:hasPropertyValue`.

SAREF extensions should not define sub-classes of `saref:PropertyValue`, as this would duplicate information already part of the taxonomy of properties.

5.6.5 Specializing OPs and DPs related to properties

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:hasProperty
- saref:isPropertyOf
- saref:hasPropertyOfInterest
- saref:isPropertyOfInterestOf
- saref:hasPropertyKind
- saref:hasPropertyValue
- saref:isValueOfProperty
- saref:hasValue
- saref:isMeasuredIn

NOTE: In some SAREF extensions or application, it may be specified that features of interest of a specific category can have only one property of a certain kind. Cardinalities different than zero or one should not be defined using this type of axioms.

EXAMPLE:
```
ex:WaterDevice a owl:Class ;
                    rdfs:subClassOf saref:Device , [
                      owl:onProperty saref:hasProperty ;
                      owl:cardinality 1 ;
                      owl:onClass [
                        owl:onProperty saref:hasPropertyKind ;
                        owl:hasValue ex:HotInletWaterTemperature ] ] .
```

5.7 SAREF pattern for states

5.7.1 Introduction

In SAREF, states refer to the identifiable conditions that features of interest are or may be in, and that can be acted upon by devices, such as observed and controlled. While states can apply to different features of interest, states of interest are specific to a feature of interest.

Figure 6 illustrates the main classes and properties in the SAREF Core patterns for States.

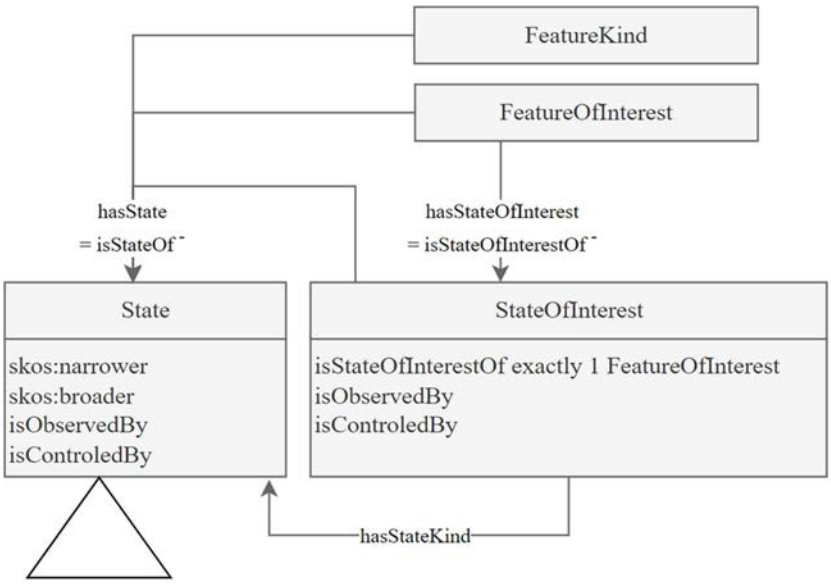

Figure 6: SAREF Core pattern for States: states and states of interest

5.7.2 States

5.7.2.1 Creating states

SAREF extensions should create instances of the class `saref:State`, such that they can be reused in different SAREF applications.

SAREF applications may also create specific instances of the class `saref:State`.

The identifier and the label of an instance of state should not contain "state" and its comment may mention "state kind" or "kind of state".

EXAMPLE:
```
ex:Open a saref:State ;
  rdfs:label "Open"@en ;
  rdfs:comment "The situation of being in an open state. A kind
of state"@en .
```

5.7.2.2 Specializing states

SAREF extensions may define a taxonomy of states using `skos:broader and skos:narrower`.

The identifier, the label, and the comment of an instance of `saref:State` should explicit its position in the taxonomy of states.

NOTE: Depending on the use case, the state can explicit the physical dimension of the property it relates to (ex. `ex:ColdTemperature`), the feature of interest it applies to (ex. `ex:AlarmActivated`), the individual states kinds below in the taxonomy (ex. `ex:OpenClose`), etc.

EXAMPLE:
```
ex:Open, is a specialization of ex:OpenClose.
ex:Open a saref:State ;
  skos:broader ex:OpenClose ;
  rdfs:label "Open"@en ;
  rdfs:comment "The situation of being in an open state. A kind
of state."@en.
```

5.7.2.3 Categorizing states

SAREF extensions and applications may define categories of states by creating sub-classes of `saref:State`.

The identifier, the label, and the comment of a sub-class of `saref:State` should explicit its position in the hierarchy of state classes.

The identifier, and the label of a state class should end with "State", and its comment should mention "state class" or "class of states".

EXAMPLE: The class `ex:WashingState` groups states related to the phases in which a washing device may be in.

```
ex:WashingState a owl:Class ;
   rdfs:subClassOf saref:State ;
   rdfs:label "Washing State"@en ;
   rdfs:comment "Class of states related to the washing
phases."@en .
```

5.7.3 States of interest

5.7.3.1 Creating states of interest

SAREF extensions should not create specific instances of the class `saref:StateOfInterest`, as they are meant to be specific to an application. The only exceptions to this provision are if the associated feature of interest is intended to be used by most applications of the SAREF extension.

Given a state of interest belongs to exactly one feature of interest, its identifier should consist of the identifier of the feature of interest, followed by character '#' and a fragment identifier. The fragment identifier part of the IRI of a state of interest should not contain "state".

A specific instance of the class `saref:StateOfInterest` should link to its kind (an instance of `saref:State`) using property `saref:hasStateKind`, and to the unique feature of interest it is a state of using OP `saref:isStateOfInterestOf`.

5.7.3.2 Categorizing states of interest

SAREF extensions should not define sub-classes of `saref:StateOfInterest`, as this would duplicate information already part of the taxonomy of states.

5.7.4 Specializing OPs and DPs related to states

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:hasState
- saref:isStateOf
- saref:hasStateOfInterest
- saref:isStateOfInterestOf
- saref:hasStateKind

NOTE: In some SAREF extensions or application, it may be specified that features of interest of a specific category can have only one state of a certain kind. Cardinalities different than zero or one should not be defined using this type of axioms.

EXAMPLE:
```
ex:WaterDevice a owl:Class ;
                    rdfs:subClassOf saref:Device , [
                      owl:onProperty saref:hasState ;
                      owl:cardinality 1 ;
                      owl:onClass [
                        owl:onProperty saref:hasStateKind ;
                        owl:hasValue ex:HotInletOpenClose ] ] .
```

5.8 SAREF pattern for functions

5.8.1 Introduction

In SAREF, functions are logical groups of commands that devices support to accomplish their tasks. Function can act upon (OP `saref:actsUpon` and its sub-properties) features, properties or states. While functions are independent of any devices, functions of interest are functions actually supported by a device.

Figure 7 illustrates the main classes and properties in the SAREF Core patterns for Functions.

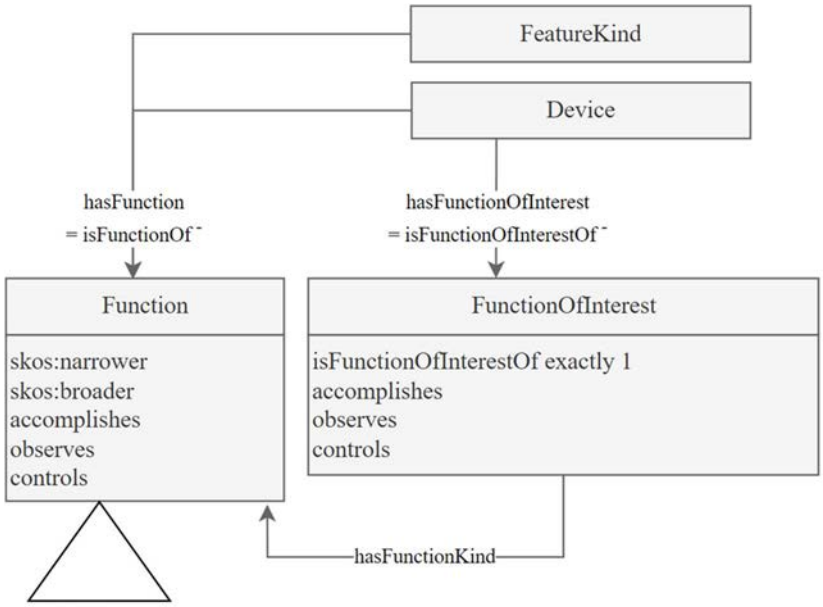

Figure 7: SAREF Core pattern for Functions: functions and functions of interest

5.8.2 Functions

5.8.2.1 Creating functions

SAREF extensions should create instances of the class `saref:Function`, such that they can be reused in different SAREF applications.

SAREF applications may also create specific instances of the class `saref:Function`.

The identifier and the label of an instance of function should end with "Function" and its comment may mention "function kind" or "kind of function".

EXAMPLE:
```
ex:OnOffControlFunction a saref:Function ;
     rdfs:label "on off function"@en ;
     rdfs:comment "The on off function represents the
functionality to control the on/off state of a feature of
interest. The on off function has three optional commands to turn
on, to turn off, and to toggle the state."@en ;
     saref:controls ex:OnOff .
```

5.8.2.2 Specializing functions

SAREF extensions may define a taxonomy of functions using `skos:broader and skos:narrower`.

The identifier, the label, and the comment of an instance of `saref:Function` should explicit its position in the taxonomy of functions.

NOTE: The set of functions is specialized down the taxonomy of functions. The observed or controlled property or state may be specialized, individual commands may be specialized, new commands may be introduced, optional commands may be deleted or become mandatory, etc.

EXAMPLE: `ex:ColorLightControlFunction, may be a specialization of ex:LightControlFunction.`

5.8.2.3 Categorizing functions

SAREF extensions and applications may define categories of functions by creating sub-classes of `saref:Function`.

The identifier, the label, and the comment of a sub-class of `saref:Function` should explicit its position in the hierarchy of function classes.

The identifier, and the label of a function class should end with "FunctionCategory", and its comment should mention "function class" or "class of functions".

EXAMPLE: The class `ex:LightingFunctionCategory` groups functions related to the lighting task.

```
ex:LightingFunctionCategory a owl:Class ;
  rdfs:subClassOf saref:Function, [
    a owl:Restriction ;
    owl:onProperty saref:accomplishes ;
    owl:hasValue ex:Lighting ] ;
  rdfs:label "Lighting Function Category"@en ;
  rdfs:comment "Category of functions that accomplish
lighting."@en .
```

5.8.3 Functions of interest

5.8.3.1 Creating functions of interest

SAREF extensions should not create specific instances of the class `saref:FunctionOfInterest`, as they are meant to be specific to an application. The only exceptions to this provision are if the associated device is intended to be used by most applications of the SAREF extension.

Given a function of interest belongs to exactly one device, its identifier should consist of the identifier of the device, followed by character '#' and a fragment identifier. The fragment identifier part of the IRI of a state of interest should not contain "function".

A specific instance of the class `saref:FunctionOfInterest` should link to its kind (an instance of `saref:Function`) using property `saref:hasFunctionKind`, and to the unique device it is a function of using OP `saref:isFunctionOfInterestOf`.

5.8.3.2 Categorizing functions of interest

SAREF extensions should not define sub-classes of `saref:FunctionOfInterest`, as this would duplicate information already part of the taxonomy of functions.

5.8.4 Specializing OPs and DPs related to functions

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:hasFunction
- saref:isFunctionOf
- saref:hasFunctionOfInterest
- saref:isFunctionOfInterestOf
- saref:hasFunctionKind

5.9 SAREF pattern for commands

5.9.1 Introduction

In SAREF, commands represent the lowest-level directives a device supports and exposes to some network. Commands can act upon (OP `saref:actsUpon` and its sub-properties) features, properties, or states. While commands are independent of any function, commands of interest are commands actually supported by a function of interest.

Figure 8 illustrates the main classes and properties in the SAREF Core patterns for Commands.

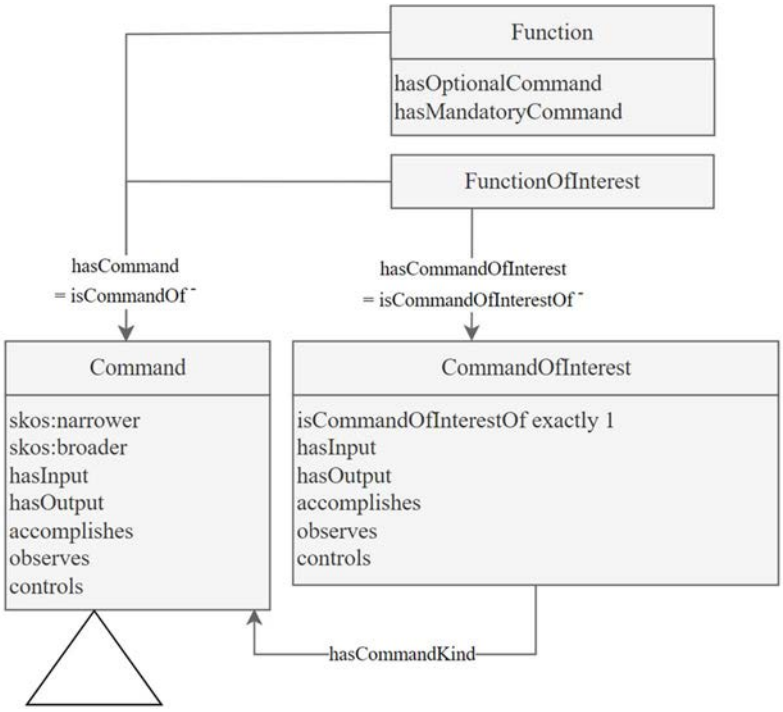

Figure 8: SAREF Core pattern for Commands: commands and commands of interest

5.9.2 Commands

5.9.2.1 Creating commands

SAREF extensions should create instances of the class `saref:Command`, such that they can be reused in different SAREF applications.

SAREF applications may also create specific instances of the class `saref:Command`.

The identifier and the label of an instance of command should end with "Command", and its comment may mention "command kind" or "kind of command".

EXAMPLE:
```
ex:OpenCommand a saref:Command ;
  rdfs:label "Open command"@en ;
  rdfs:comment "A command to put some feature of interest in an
open state. This command has no input or output parameter."@en ;
  saref:controls ex:OpenClose .
```

5.9.2.2 Specializing commands

SAREF extensions may define a taxonomy of commands using `skos:broader and skos:narrower`.

The identifier, the label, and the comment of an instance of `saref:Command` should explicit its position in the taxonomy of commands.

NOTE: The set of commands is specialized down the taxonomy of functions. The observed or controlled property or state may be specialized, input and output may be added, specialized, etc.

EXAMPLE: `ex:OffCommand`, may be a specialization of `ex:OnOffCommand`.

5.9.2.3 Categorizing commands

SAREF extensions and applications may define categories of commands by creating sub-classes of `saref:Command`.

The identifier, the label and the comment of a sub-class of `saref:Command` should explicit its position in the hierarchy of function classes.

The identifier, and the label of a command class should end with "CommandCategory", and its comment should mention "command category" or "category of commands".

EXAMPLE: The class `ex:TemperatureInputCommandCategory groups commands that take the ex:Temperature property as input.`
```
ex:TemperatureAsInputCommandCategory a owl:Class ;
  rdfs:subClassOf saref:Command, [
    a owl:Restriction ;
    owl:onProperty saref:hasInput;
    owl:hasValue ex:Temperature ] ;
  rdfs:label "Temperature As Input Command Category"@en ;
  rdfs:comment "Category of commands that take a temperature as
input."@en .
```

5.9.3 Commands of interest

5.9.3.1 Creating commands of interest

SAREF extensions should not create specific instances of the class `saref:CommandOfInterest`, as they are meant to be specific to an application. The only exception to this provision is if the associated function of interest is intended to be used by most applications of the SAREF extension.

Given a command of interest belongs to exactly one function, itself belonging to exactly one device, its identifier should consist of the identifier of the function, followed by character '/' and a local identifier for the command.

A specific instance of the class `saref:CommandOfInterest` should link to its kind (an instance of `saref:Command`) using property `saref:hasCommandKind`, and to the unique function of interest it is a command of using OP `saref:isCommandOfInterestOf`.

5.9.3.2 Categorizing commands of interest

SAREF extensions should not define sub-classes of `saref:CommandOfInterest`, as this would duplicate information already part of the taxonomy of commands.

5.9.4 Specializing OPs and DPs related to commands

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:hasCommand
- saref:isCommandOf
- saref:hasMandatoryCommand
- saref:hasOptionalCommand
- saref:hasCommandOfInterest
- saref:isCommandOfInterestOf
- saref:hasCommandKind

SAREF extensions and applications may define sub-properties of OP `saref:hasInput`. The identifier, the label, and the comment, of such a sub-property should contain "Input".

EXAMPLE 1: `ex:hasTargetColorInput` and `ex:hasTransitionTimeInput`.

SAREF extensions and applications may define sub-properties of OP `saref:hasOutput`. The identifier, the label, and the comment, of such a sub-property should contain with "Output".

EXAMPLE 2: `ex:hasOutputTemperature, ex:hasOutputHumidity`.

5.10 SAREF pattern for services and operations

A `saref:Service` is a digital representation of a function in a network, making it discoverable, registerable and remotely controllable in the network.

A `saref:Operation` is the means of a service to communicate in a procedure-type manner over the network (i.e. transmit data to/from other devices). It is the -machine interpretable- exposure of a -human understandable- command to a network.

Figure 9 illustrates the main classes and properties in the SAREF Core patterns for Services and Operations.

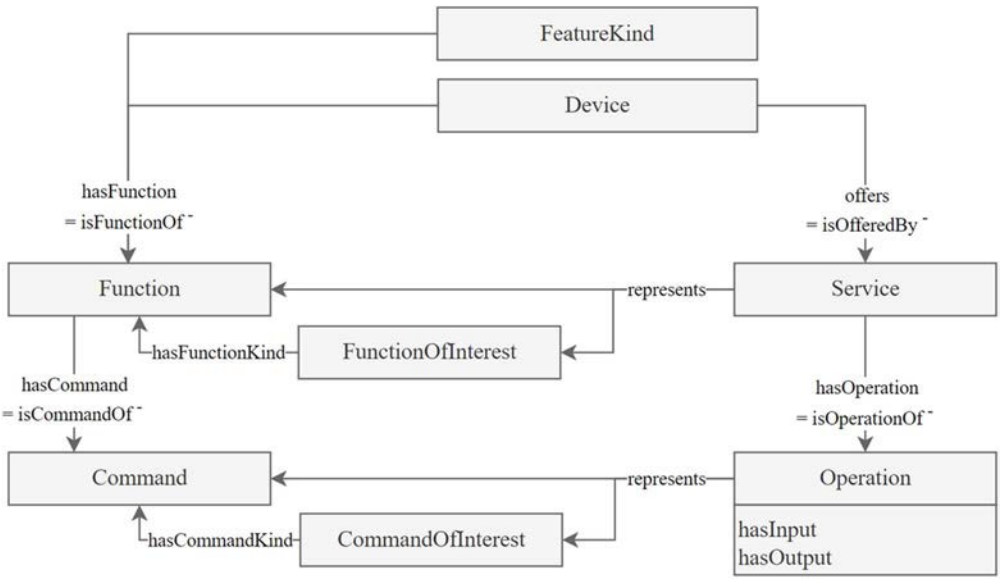

Figure 9: SAREF Core pattern for Services and Operations

The present document does not constrain the usage of the classes services and operations.

NOTE: This may change in a future release of the present document, after enough implementation evidence of these concepts is collected.

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:offers
- saref:isOfferedBy
- saref:hasOperation
- saref:isOperationOf
- saref:represents

5.11 SAREF pattern for procedure executions

A `saref:ProcedureExecution` represents the act of carrying out a procedure.

SAREF Core defines four sub-classes of `saref:ProcedureExecution`:

- `saref:CommandExecution` describes the execution of a command.
- `saref:OperationExecution` describes the execution of an operation in a network: the-machine interpretable- description of a communication between devices over the network.
- `saref:Observation` is the act of carrying out a procedure to estimate or calculate a value of a property of a feature of interest, or a state of a feature of interest.
- `saref:Actuation` is the act of carrying out a procedure to control the state of the world using an actuator. It links to an actuator to describe what made the actuation, and to the controlled feature, property, property of interest, state, or state of interest.

Figure 10 illustrates the main classes and properties in the SAREF Core patterns for procedure executions.

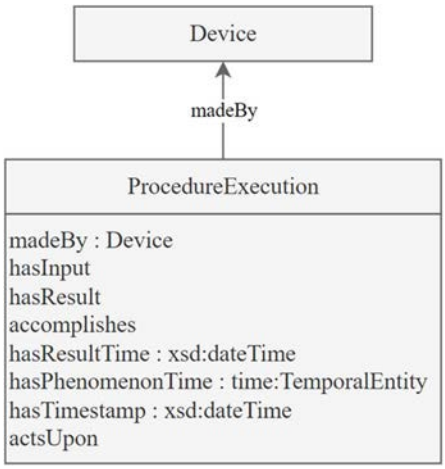

Figure 10: SAREF Core pattern for Procedure Executions

SAREF extensions should not create specific instances of the class `saref:ProcedureExecution` or any of its sub-classes, as they are meant to be created by applications.

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:madeBy
- saref:isMadeBy
- saref:hasResultTime

- saref:hasPhenomenonTime
- saref:hasTimestamp

5.12 SAREF pattern for profiles

A device in SAREF can be characterized by profiles. A `saref:Profile` describes the money earned (negative values) or paid (positive values) for the use (production or consumption) of a commodity by a device in a certain context.

Figure 11 illustrates the main classes and properties in the SAREF Core patterns for Profiles.

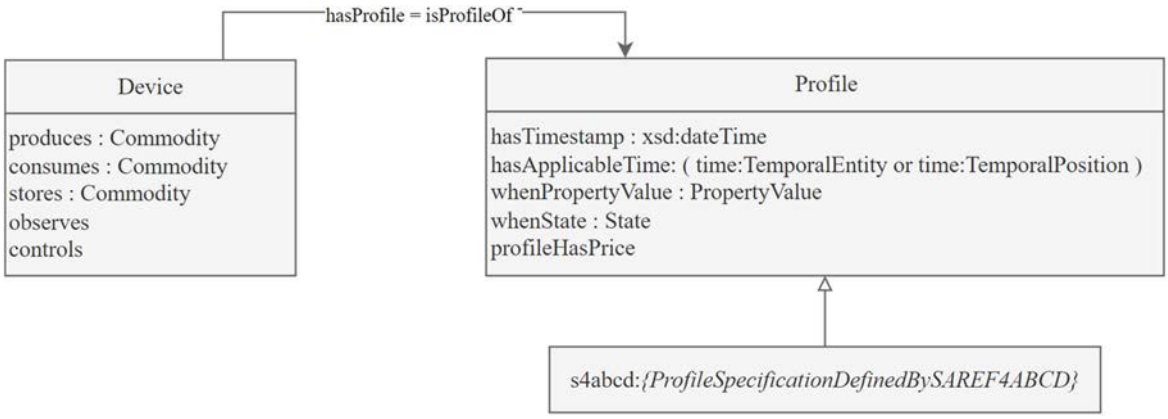

Figure 11: SAREF Core pattern for Profiles

SAREF extensions should not create specific instances of the class `saref:Profile` or any of its sub-classes, as they are meant to be created by applications.

SAREF extensions and applications shall not define sub-properties of the following properties:

- saref:hasProfile
- saref:isProfileOf
- saref:whenPropertyValue
- saref:whenState
- saref:profileHasPrice

6 SAREF Patterns for Systems, Connections and Connection Points

6.1 Ontology and semantics

6.1.1 Introduction

The present clause is the technical specification of SAREF4SYST, a generic extension of ETSI TS 103 264 [1] SAREF to represent the topology of systems and how they are connected or interact.

SAREF4SYST defines Systems, Connections between systems, and Connection Points at which systems may be connected. These core concepts can be used generically to define the topology of features of interest, which is highly important in many use cases. If a room holds a lighting device, and if it is adjacent with an open window to a room whose luminosity is low, then by turning on the lighting device in the former room one may expect that the luminosity in the latter room will rise.

The SAREF4SYST ontology pattern can be applied to different domains. For example to describe zones inside a building (systems), that share a frontier (connections). Properties of systems are typically state variables (e.g. agent population, temperature), whereas properties of connections are typically flows (e.g. heat flow).

The core ontology for SAREF4SYST is a lightweight OWL-DL ontology that defines 3 classes and 9 object properties.

Use cases for ontology patterns are described extensively in ETSI TR 103 549 [i.4]. Clauses 6.1.2 and 6.1.3 extract use cases for the SAREF4SYST ontology pattern.

6.1.2 Application example 1: Smart Energy

The present clause illustrates how SAREF4SYST can be used to homogeneously represent knowledge that is relevant for use cases in the Smart Energy domain:

- Electric power systems can exchange electricity with other electric power systems. The electric energy can flow both ways in some cases (from the Public Grid to a Prosumer), or in only one way (from the Public Grid to a Load). Electric power systems can be made up of different sub-systems. Generic sub-types of electric power systems include producers, consumers, storage systems, transmission systems.
- Electric power systems may be connected one to another through electrical connection points. An Electric power system may have multiple connection points (Multiple Winding Transformer generally have one single primary winding with two or more secondary windings). Generic sub-types of electrical connection points include plugs, sockets, direct-current, single-phase, three-phase, connection points.
- An Electrical connection may exist between two Electric power systems at two of their respective connection points. Generic sub-types of electrical connections include Single-phase Buses, Three-phase Buses. A single-phase electric power system can be connected using different configurations at a three-phase bus (RN, SN, TN types).

6.1.3 Application example 2: Smart Building

The present clause illustrates how SAREF4SYST can be used to homogeneously represent knowledge that is relevant for use cases in the Smart Building domain:

- Buildings, Storeys, Spaces, are different sub-types of Zones. Zones can contain sub-zones. Zones can be adjacent or intersect with other zones.
- Two zones may share one or more connections. For example some fresh air may be created inside a storey if it has two controllable openings to the exterior at different cardinal points.

6.1.4 Namespaces

In addition to prefixes and namespaces defined in Table 1, clause 5 uses those in Table 2.

Table 2: Prefixes and namespaces used in clause 5

Prefix	Namespace
s4syst-ex	https://saref.etsi.org/saref4syst/v1.1.2/example/example/

6.2 Core Ontology

6.2.1 General overview

A graphical overview of the SAREF4SYST ontology is provided in Figure 12. Four pairs of properties are inverse one of the other; the property `s4syst:connectedTo` is symmetric, and properties `s4syst:hasSubSystem` and `s4syst:hasSubSystem` are transitive.

Clauses 6.2.2 to 6.2.4 describe the different parts of the SAREF4SYST core ontology describing the different conceptual modules of the ontology.

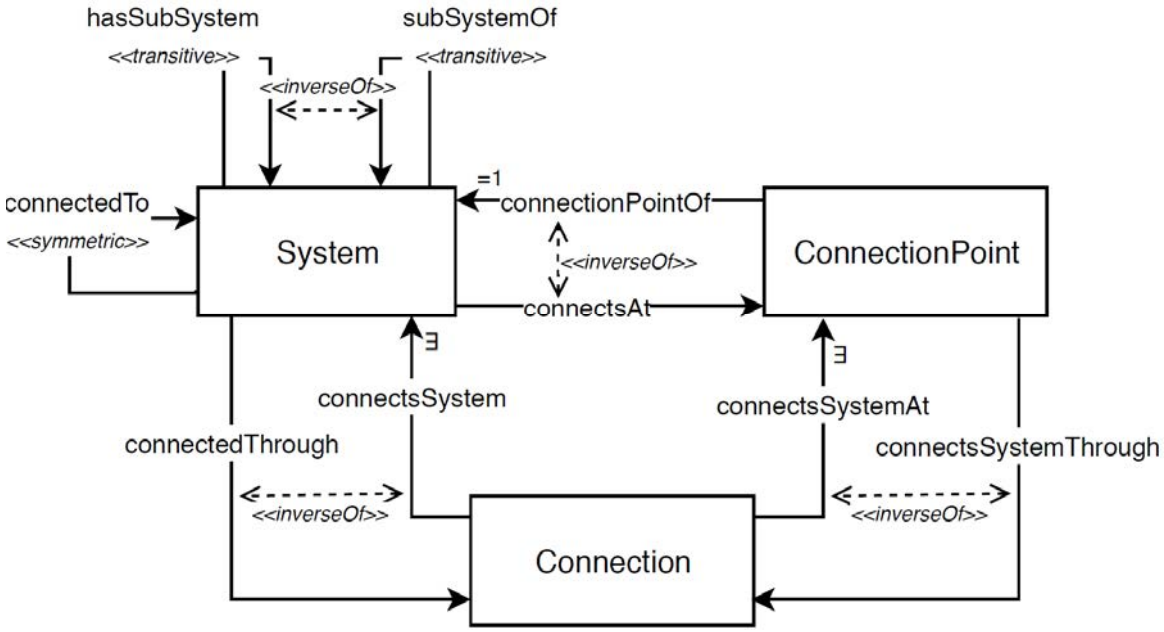

Figure 12: SAREF4SYST overview

6.2.2 Systems and sub-systems

A `s4syst:System`, is defined as a part of the universe that is virtually isolated from the environment.

NOTE: The system properties are typically state variables (e.g. consumed or stored energy, agent population, temperature, volume, humidity).

Figure 13 illustrates classes and properties that can be used to define connected systems and their sub-systems.

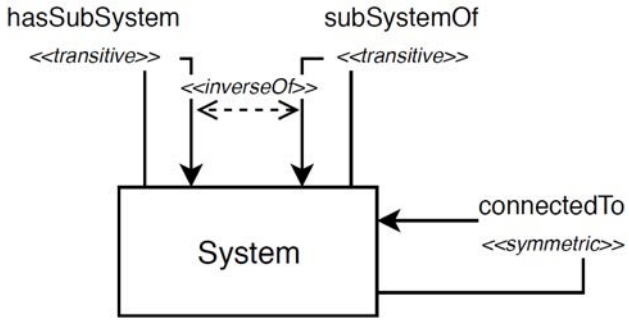

Figure 13: SAREF4SYST: Systems, sub-systems

A system may be connected to other systems that are part of its environment. This is modelled by a property named `s4syst:connectedTo`, which is symmetric.

EXAMPLE 1: `<electric_vehicle> s4syst:connectedTo <electric_vehicle_service_equipment>.`

Connected systems interact in some ways. The exact meaning of interact is defined by sub-properties of `s4syst:connectedTo`.

EXAMPLE 2: For the electricity to directly flow between an electric vehicle service equipment `<electric_vehicle_service_equipment>` and an electric vehicle `<electric_vehicle>`, then they should be linked by property `s4syst-ex:exchangesElectricityWith`:.

`<electric_vehicle> s4syst-ex:exchangesElectricityWith <electric_vehicle_service_equipment>.`

A system can be a sub-system of another system. This is modelled using the transitive properties `s4syst:subSystemOf` and `s4syst:hasSubSystem`.

EXAMPLE 3: `<battery> s4syst:subSystemOf <electric_vehicle>.`

Properties of subsystems somehow contribute to the properties of the super system. The exact meaning of this contribution is defined by sub properties of `s4syst:subSystemOf`.

EXAMPLE 4: If one wants to model the fact that the consumption power of a fridge `<fridge/1>` contributes to the consumption power of the kitchen, `<kitchen/1>`, then one may use a sub-property of `s4syst:subSystemOf` named `s4syst-ex:subElectricPowerSystemOf`.

`<fridge/1> s4syst-ex:subElectricPowerSystemOf <kitchen/1>.`

Table 3 summarizes the restrictions that characterize the `s4syst:hasSubSystem` property.

Table 3: Restrictions of the `s4syst:hasSubSystem` property

Axiom	Definition
Domain: `s4syst:System`	The `s4syst:hasSubSystem` connects only `s4syst:Systems`.
Range: `s4syst:System`	The `s4syst:hasSubSystem` connects only to `s4syst:Systems`.
InverseOf `s4syst:subSystemOf`	If a `s4syst:System` has for sub-system another `s4syst:system`, then the latter is a sub-system of the former.
Transitive	The sub-system of a sub-system is a sub-system.

Table 4 summarizes the restrictions that characterize the `s4syst:subSystemOf` property.

Table 4: Restrictions of the `s4syst:subSystemOf` property

Axiom	Definition
Domain: `s4syst:System`	The `s4syst:subSystemOf` connects only `s4syst:Systems`.
Range: `s4syst:System`	The `s4syst:subSystemOf` connects only to `s4syst:Systems`.
InverseOf `s4syst:hasSystem`	If a `s4syst:System` is a sub-system another `s4syst:System`, then the latter has for sub-system the former.
Transitive	The super-system of a super-system is a super-system.

Table 5 summarizes the restrictions that characterize the `s4syst:connectedTo` property.

Table 5: Restrictions of the `s4syst:connectedTo` property

Axiom	Definition
Domain: `s4syst:System`	The `s4syst:connectedTo` connects only `s4syst:Systems.`
Range: `s4syst:System`	The `s4syst:connectedTo` connects only to `s4syst:Systems.`
Symmetric	If a `s4syst:System` is connected to another, then the latter is connected to the former.

6.2.3 Connections between systems

A connection between two `s4syst:System`s, modelled by `s4syst:connectedTo`, describes the potential interactions between connected `s4syst:System`s. A connection can be qualified using class `s4syst:Connection`.

EXAMPLE 1: One can associate a `s4syst:Connection` with properties (`saref:Property`) that describe the interactions between the connected `s4syst:System`s (e.g. population flow, exchange surface, contact temperature).

Figure 14 illustrates classes and properties that can be used to qualify connections between `s4syst:Systems`.

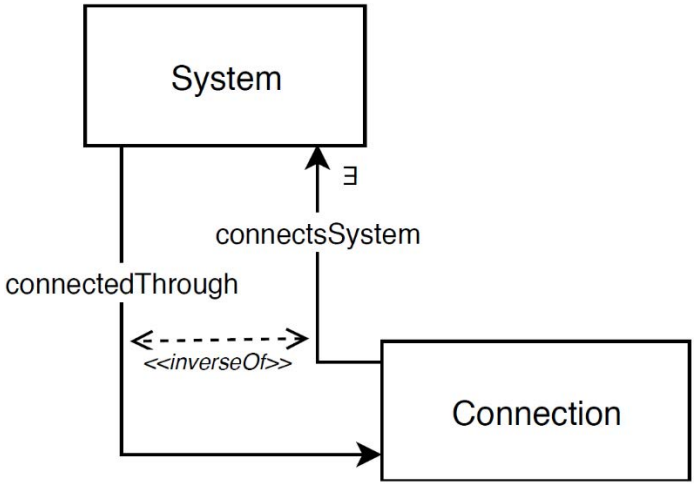

Figure 14: SAREF4SYST: Connections between systems

EXAMPLE 2: A power connection between power systems describes the fact that these systems may exchange electricity.

```
<connection> s4syst:connectsSystem <electric_vehicle> ,
<electric_vehicle_service_equipment>.
<electric_vehicle> s4syst:connectedThrough <connection>.
<electric_vehicle_service_equipment> s4syst:connectedThrough
<connection>.
```

Table 6 summarizes restrictions that characterize a `s4syst:System` with respect to connection.

Table 6: Restrictions of the `s4syst:System` class

Axiom	Definition
DisjointWith `s4syst:Connection`	No individual can be both a `s4syst:System` and a `s4syst:Connection`.

Table 7 summarizes the restrictions that characterize the `s4syst:Connection` class.

Table 7: Restrictions of the `s4syst:Connection` class

Axiom	Definition
DisjointWith `s4syst:System`	No individual can be both a `s4syst:Connection` and a `s4syst:System`.
SubClassOf `s4syst:connectsSystem` some `s4syst:System`	For any `s4syst:Connection` there exists a `s4syst:System` that it connects.

Table 8 summarizes the restrictions that characterize the `s4syst:connectsSystem` property.

Table 8: Restrictions of the `s4syst:connectsSystem` property

Axiom	Definition
Domain: `s4syst:Connection`	The `s4syst:connectsSystem` connects only `s4syst:Connections`.
Range: `s4syst:System`	The `s4syst:connectsSystem` connects only to `s4syst:Systems`.
InverseOf `s4syst:connectedThrough`	If a `s4syst:Connection` connects a `s4syst:System`, then the latter is connected through the former.

Table 9 summarizes the restrictions that characterize the `s4syst:connectedThrough` property.

Table 9: Restrictions of the `s4syst:connectedThrough` property

Axiom	Definition
Domain: `s4syst:System`	The `s4syst:connectedThrough` connects only `s4syst:Systems`.
Range: `s4syst:Connection`	The `s4syst:connectedThrough` connects only to `s4syst:Connections`.
InverseOf `s4syst:connectsSystem`	If a `s4syst:System` is connected through a `s4syst:Connection`, then the latter connects the former.

6.2.4 Connection Points of systems

A `s4syst:System` connects to other `s4syst:System`s at connection points. A connection point belongs to one and only one `s4syst:System`, and can be described using the class `s4syst:ConnectionPoint`.

Figure 15 illustrates the classes and the properties that can be used to describe connection points of a `s4syst:System`.

EXAMPLE: An electric vehicle charging station may have three `s4syst:connectionPoints`: two plugs of different kind to which electric vehicles can connect, and a three-phase connection point to the public grid:
`<electric_vehicle> s4syst:connectsAt <plug_high_voltage> , <normal_plug> , <three_phase_connection_point>.`

One can then associate a `s4syst:ConnectionPoint` with properties (`saref:Property`) that describe it (e.g. position and speed, voltage and intensity, thermic transmission coefficient).

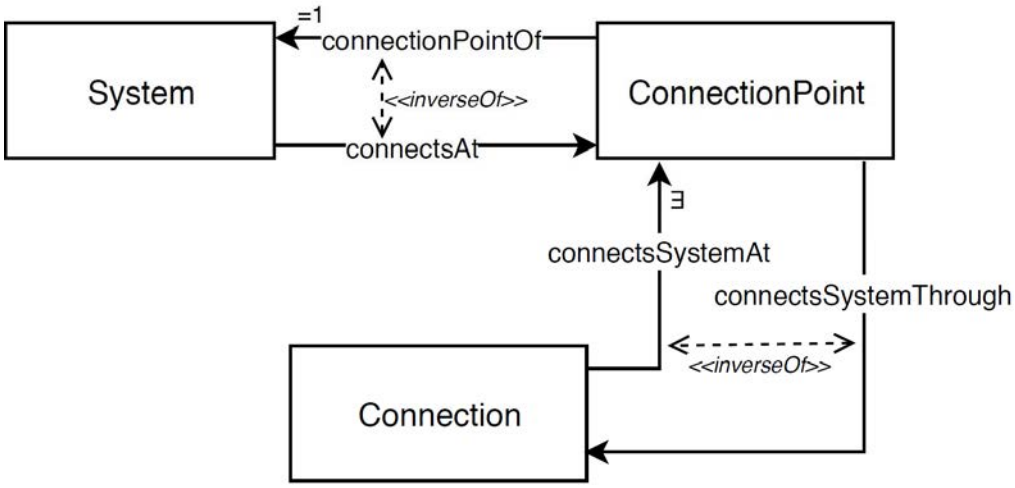

Figure 15: SAREF4SYST: Connection points of systems, where other systems connect

Table 10 summarizes additional restrictions that characterize the `s4syst:System` class with respect to connection points.

Table 10: Additional restrictions of the `s4syst:System` class

Axiom	Definition
DisjointWith `s4syst:ConnectionPoint`	No individual can be both a `s4syst:System` and a `s4syst:ConnectionPoint`.

Table 11 summarizes additional restrictions that characterize the `s4syst:Connection` class with respect to `s4syst:ConnectionPoints`.

Table 11: Restrictions of the `s4syst:Connection` class

Axiom	Definition
DisjointWith `s4syst:ConnectionPoint`	No individual can be both a `s4syst:Connection` and a `s4syst:ConnectionPoint`.
SubClassOf `s4syst:connectsSystemAt` some `s4syst:ConnectionPoint`	For any `s4syst:Connection` there exists a `s4syst:ConnectionPoint` that it connects a system at.

Table 12 summarizes restrictions that characterize the `s4syst:ConnectionPoint` class.

Table 12: Restrictions of the `s4syst:ConnectionPoint` class

Axiom	Definition
DisjointWith `s4syst:System`	No individual can be both a `s4syst:System` and a `s4syst:ConnectionPoint`.
DisjointWith `s4syst:Connection`	No individual can be both a `s4syst:Connection` and a `s4syst:ConnectionPoint`.
SubClassOf `s4syst:connectionPointOf` exactly 1 `owl:Thing`	A `s4syst:ConnectionPoint` is always the `s4syst:ConnectionPoint` of something (some `s4syst:System`).

Table 13 summarizes restrictions that characterize the `s4syst:connectedThrough` property.

Table 13: Restrictions of the `s4syst:connectionPointOf` property

Axiom	Definition
Domain: `s4syst:ConnectionPoint`	Only `s4syst:ConnectionPoints` may be subject of a `s4syst:connectionPointOf` property.
Range: `s4syst:System`	Only `s4syst:systems` may be object of a `s4syst:connectionPointOf` property.
Functional	A `s4syst:ConnectionPoint` may be the `s4syst:connectionPoint` of only one `s4syst:System`.
InverseOf: `s4syst:connectsAt`	If a `s4syst:ConnectionPoint` is a `s4syst:ConnectionPoint` of a `s4syst:System`, then the latter connects at the former.

Table 14 summarizes restrictions that characterize the `s4syst:connectsAt` property.

Table 14: Restrictions of the `s4syst:connectsAt` property

Axiom	Definition
Domain: `s4syst:System`	Only `s4syst:Systems` may be subject of a `s4syst:connectsAt` property.
Range: `s4syst:ConnectionPoint`	Only `s4syst:ConnectionPoint` may be object of a `s4syst:connectsAt` property.
InverseOf: `s4syst:connectionPointOf`	If a `s4syst:System` connects at a `s4syst:ConnectionPoint`, then the latter is a `s4syst:ConnectionPoint` of the former.

Table 15 summarizes restrictions that characterize the s4syst:connectSystemThrough property.

Table 15: Restrictions of the s4syst:connectSystemThrough property

Axiom	Definition
Domain: s4syst:ConnectionPoint	Only s4syst:ConnectionPoints may be subject of a s4syst:connectsSystemThrough property.
Range: s4syst:Connection	Only s4syst:Connections may be object of a s4syst:connectsSystemThrough property.
InverseOf: s4syst:connectsSystemAt	If a s4syst:ConnectionPoints connects a s4syst:System through a s4syst:Connection, then the latter connects the s4syst:System at the former.

Table 16 summarizes restrictions that characterize the s4syst:connectsSystemAt property.

Table 16: Restrictions of the s4syst:connectsSystemAt property

Axiom	Definition
Domain: s4syst:Connection	Only s4syst:Connections may be subject of a s4syst:connectsSystemAt property.
Range: s4syst:ConnectionPoint	Only s4syst:ConnectionPoints may be object of a s4syst:connectsSystemAt property.
InverseOf: s4syst:connectSystemThrough	If a s4syst:Connection connects a s4syst:System at a s4syst:ConnectionPoint, then the latter connects the s4syst:System through the former.

Table 17 summarizes property chain axioms that characterize the SAREF4SYST properties.

Table 17: Property chain axioms on the SAREF4SYST properties

Axiom	Definition
Property chain s4syst:connectsAt o s4syst:connectsSystemThrough is sub-property of s4syst:connectedThrough	If a s4syst:System connects at one of its s4syst:ConnectionPoint to some s4syst:Connection, then the s4syst:System connects through this s4syst:Connection.
Property chain s4syst:connectsSystemAt o s4syst:connectionPointOf is sub-property of s4syst:connectsSystem	If a s4syst:Connection connects a s4syst:ConnectionPoint of a s4syst:System, then it connects that s4syst:System.

6.3 Pattern application to verticals

6.3.1 Introduction

Applications of the SAREF4SYST pattern are subsets of ontologies that define sub-classes and/or sub-properties of the classes and properties defined in clause 6. SAREF or SAREF extensions should contain applications of the SAREF4SYST pattern.

Clauses 6.3.2 to 6.3.4 define how such applications are made.

6.3.2 Systems and sub-systems

6.3.2.1 Sub-classes of s4syst:System

A specialization of the SAREF4SYST pattern may define a sub-class of the s4syst:System class. If defined, it shall have an English-tagged label (rdfs:label) ending with "System". The local name of its IRI shall be a camel case form of its English-tagged label.

EXAMPLE: `s4syst-ex:ElectricPowerSystem` has English-tagged label `"Electric Power System".@en.`

It shall have an English-tagged comment (`rdfs:comment`) that defines it in natural language.

6.3.2.2 Sub-properties of `s4syst:hasSubSystem and s4syst:subSystemOf`

A specialization of the SAREF4SYST pattern may define a sub-property of the `s4syst:hasSubSystem` property or the `s4syst:subSystemOf` property. If defined, it shall have an English-tagged label (`rdfs:label`), and the local name of its IRI shall be a mixed case form of its English-tagged label. The label of the sub-property shall use the same morpho-syntactic structure as its super-property.

EXAMPLE 1: `s4syst-ex:subElectricPowerSystemOf` with English-tagged label `"sub electric power system of".@en.`

A sub-class of `s4syst:System` shall have an English-tagged comment (`rdfs:comment`) that describes in natural language how the properties of the sub-system contribute to the properties of the super-system.

EXAMPLE 2: `s4syst-ex: subElectricPowerSystemOf` has English-tagged comment that contains `"The consumed electricity of an electric power consumer system contributes to the consumed electricity of its super electric power consumer system".`

A sub-class of `s4syst:System` may be defined as a sub-class or equivalent class of an anonymous class having an existential or universal restriction on the `s4syst:hasSubSystem` or `s4syst:subSystemOf` property. If so, the class shall be `s4syst:System` or one of its sub-classes.

6.3.2.3 Sub-properties of `s4syst:connectedTo`

A specialization of the SAREF4SYST pattern may define a sub-property of the `s4syst:connectedTo` property. If defined, it shall have an English-tagged label (`rdfs:label`), and the local name of its IRI shall be a mixed case form of its English-tagged label.

EXAMPLE 1: `s4syst-ex:exchangesElectricityWith` has English-tagged label `"exchanges electricity with".@en.`

It shall have an English-tagged comment (`rdfs:comment`) that describes in natural language how the connected `s4syst:System`s interact.

A sub-property of the `s4syst:connectedTo` property may be symmetric. If so, it shall define a common domain and range that is `s4syst:System` or one of its sub-classes. The English-tagged comment shall reflect this symmetry. The comment shall mention the domain and range `s4syst:System`s.

EXAMPLE 2: `s4syst-ex:exchangesElectricityWith` has English-tagged comment `"Links an electric power system to another electric power system with which it may exchange electricity".@en.`

If a sub-property of the `s4syst:connectedTo` property is not symmetric, then an inverse (`owl:inverseOf`) of this property may be defined. If defined, the English-tagged labels and comments of these two properties shall reflect this aspect.

EXAMPLE 3: `s4syst-ex:powers` has English-tagged comment `"Links an electric power system to another electric power system to which it sends electricity".@en,` and `s4syst-ex:isPoweredBy` has English-tagged comment `"Links an electric power system to another electric power system from which it receives electricity".@en.`

6.3.3 Connections between systems

6.3.3.1 Sub-classes of `s4syst:Connection`

A specialization of the SAREF4SYST pattern may define a sub-class of the `s4syst:Connection` class. If defined, it shall have an English-tagged label (`rdfs:label`) ending with "Connection". The local name of its IRI shall be a camel case form of its English-tagged label.

EXAMPLE 1: `s4syst-ex:ThreePhasePowerBusConnection` has English-tagged label `"Three-Phase Power Bus Connection".@en.`

A sub-class of the `s4syst:Connection` class shall have an English-tagged comment (`rdfs:comment`) that defines it in natural language.

EXAMPLE 2: `s4syst-ex:ThreePhasePowerBusConnection` has English-tagged comment `"A three-phase power bus connection is a connection between electric power systems composed of four wires (plus the protective earth): wires R, S, T, for the phases; wire N for the neutral".@en.`

A sub-class of the `s4syst:Connection` class may have a universal restriction on the property `s4syst:connectsSystem` to `s4syst:System` or one of its sub-classes. If so, then the label or the comment of the sub-class of `s4syst:Connection` and the sub-class of `s4syst:System` shall show this relation ostensibly.

EXAMPLE 3: The class `s4syst-ex:ElectricalConnection` has a universal restriction on the property `s4syst:connectsSystem` to `seas-ex:ElectricPowerSystem.`

A sub-class of the `s4syst:Connection` class may be defined as disjoint from other sub-classes of `s4syst:Connection`.

EXAMPLE 4: The following classes are pairwise disjoint:
`s4syst-ex:SinglePhasePowerBusConnection,`
`s4syst-ex:SplitPhasePowerBusConnection,`
`s4syst-ex:ThreePhasePowerBusConnection.`

6.3.3.2 Sub-properties of `s4syst:connectedThrough`

A specialization of the SAREF4SYST pattern may define a sub-property of the `s4syst:connectedThrough` property. If defined, it shall have an English-tagged label (`rdfs:label`), and the local name of its IRI shall be a mixed case form of its English-tagged label. The label of the sub-property should use the same morpho-syntactic structure as its super-property.

EXAMPLE 1: `s4syst-ex:connectedInStarThrough` has English-tagged label `"connected in star through".@en.`

A sub-property of the `s4syst:connectedThrough` property shall have an English-tagged comment (`rdfs:comment`) that describes in natural language how the connected `s4syst:System`s interact.

EXAMPLE 2: `s4syst-ex:connectedInStarThrough` has English-tagged comment `"Links an electric power system to a three-phase power bus connection with which it is connected with a star configuration".@en.`

A sub-property of the `s4syst:connectedThrough` property may have an inverse property. If defined, the English-tagged labels and comments of these two properties shall reflect this aspect.

EXAMPLE 3: `s4syst-ex:connectedInStarThrough` and `s4syst-ex:connectsSystemInStar` are inverse properties.

A sub-class of the `s4syst:System` class may have an existential restriction on a sub-property of `s4syst:connectedThrough` to some sub-class of `s4syst:Connection`.

EXAMPLE 4: `s4syst-ex:ElectricPowerTransformer` has existential restrictions on `s4syst-ex:primarilyConnectedThrough` and `s4syst-ex:secondarilyConnectedThrough` to `s4syst-ex:ElectricalConnection`.

A sub-property of the `s4syst:connectedThrough` property may be defined as disjoint from other sub-properties of `s4syst:connectedThrough`.

EXAMPLE 5: The following properties are disjoint: `s4syst-ex:connectsSystemInStar` and `s4syst-ex:connectsSystemInTriangle`.

6.3.3.3 Sub-properties of `s4syst:connectsSystem`

A specialization of the SAREF4SYST pattern may define a sub-property of the `s4syst:connectsSystem` property. If defined, it shall have an English-tagged label (`rdfs:label`), and the local name of its IRI shall be a mixed case form of its English-tagged label. The label of the sub-property should use the same morpho-syntactic structure as its super-property.

EXAMPLE 1: `s4syst-ex:connectsSystemInStar` has English-tagged label `"connects system in star".@en`.

A sub-property of the `s4syst:connectsSystem` property shall have an English-tagged comment (`rdfs:comment`) that describes in natural language how the connected `s4syst:system`s interact.

EXAMPLE 2: `s4syst-ex:connectsSystemInStar` has English-tagged comment `"Links a three-phase power bus to one of the electric power systems it connects with a star configuration".@en`.

A sub-property of the `s4syst:connectsSystem` property may have an inverse property. If defined, the English-tagged labels and comments of these two properties shall reflect this aspect.

EXAMPLE 3: `s4syst-ex:connectsSystemInStar` and `s4syst-ex:connectedInStarThrough are inverse properties.`

6.3.4 Connection Points of systems

6.3.4.1 Sub-classes of `s4syst:ConnectionPoint`

A specialization of the SAREF4SYST pattern may define a sub-class of the `s4syst:ConnectionPoint` class. If defined, it shall have an English-tagged label (`rdfs:label`) ending with "Connection Point". The local name of its IRI shall be a camel case form of its English-tagged label.

EXAMPLE 1: `s4syst-ex:ThreePhaseConnectionPoint` has English-tagged label `"Three-Phase Connection Point".@en`.

A sub-class of the `s4syst:ConnectionPoint` class shall have an English-tagged comment (`rdfs:comment`) that defines it in natural language.

EXAMPLE 2: `s4syst-ex:ThreePhaseConnectionPoint` has English-tagged comment `"A three-phase connection point is a connection point composed of four wires (plus the protective earth): wires R, S, T, for the phases; wire N for the neutral".@en`.

A sub-class of the `s4syst:ConnectionPoint` class may have a universal restriction on the property `s4syst:connectsSystemThrough` or one of its sub-properties to a sub-class of `s4syst:Connection`. If so, then the label or the comment of the sub-class of `s4syst:ConnectionPoint` and the sub-class of `s4syst:Connection` shall show this relation ostensibly.

A sub-class of the s4syst:ConnectionPoint class may have a universal restriction on the property s4syst:connectionPointOf to a sub-class of s4syst:System. If so, then the label or the comment of the sub-class of s4syst:ConnectionPoint and the sub-class of s4syst:System shall show this relation ostensibly.

EXAMPLE 3: s4syst-ex:IlluminableZoneFrontierConnectionPoint has a universal restriction on the property s4syst:connectsSystemThrough to the class s4syst-ex:LightTransmissionSystemConnection, has a universal restriction on the property s4syst:connectionPointOf to the class s4syst ex:IlluminableZoneSystem, and has English-tagged comment "The class of zones frontiers _on_ which one may measure/effect luminosity, and perceive brightness. Illuminable zones are surfaces such as walls, tables, sheer curtains, mirrors, windows. Light may be reflected, absorbed, and transmitted by illuminable zone frontier connection points".@en.

6.3.4.2 Sub-properties of s4syst:connectionPointOf

A specialization of the SAREF4SYST pattern may define a sub-property of the s4syst:connectionPointOf property. If defined, it shall have an English-tagged label (rdfs:label), and the local name of its IRI shall be a mixed case form of its English-tagged label. The label of the sub-property should use the same morpho-syntactic structure as its super-property.

A sub-property of the s4syst:connectionPointOf property shall have an English-tagged comment (rdfs:comment) that describes in natural language how the connected s4syst:System interact.

6.3.4.3 Sub-properties of s4syst:connectsAt

A specialization of the SAREF4SYST pattern may define a sub-property of the s4syst:connectsAt property. If defined, it shall have an English-tagged label (rdfs:label), and the local name of its IRI shall be a mixed case form of its English-tagged label. The label of the sub-property should use the same morpho-syntactic structure as its super-property.

EXAMPLE 1: s4syst-ex:connectsPrimarilyAt has English-tagged label "connects primarily at".@en.

A sub-property of the s4syst:connectsAt property shall have an English-tagged comment (rdfs:comment) that describes in natural language how the connected s4syst:System interact.

EXAMPLE 2: s4syst-ex:connectsPrimarilyAt has English-tagged comment "Links an electric power transformer system to its primary connection point".@en.

A sub-class of s4syst:System may be equivalent to an anonymous class having a universal restriction on s4syst:connectsAt or one of its sub-properties to a sub-class of s4syst:ConnectionPoint. If so, then the label or the comment of the sub-class of s4syst:System and the sub-class of s4syst:ConnectionPoint shall show this relation ostensibly.

EXAMPLE 3: s4syst-ex:USBCommunicationDeviceSystem is equivalent to an anonymous class having a universal restriction on s4syst:connectsAt to the class s4syst-ex:USBCommunicationConnectionPoint, and has English-tagged comment "The class of communication devices capable of communicating using the USB protocol".@en.

6.3.4.4 Sub-properties of s4syst:connectsSystemThrough

A specialization of the SAREF4SYST pattern may define a sub-property of the s4syst:connectsSystemThrough property. If defined, it shall have an English-tagged label (rdfs:label), and the local name of its IRI shall be a mixed case form of its English-tagged label. The label of the sub-property should use the same morpho-syntactic structure as its super-property.

EXAMPLE 1: s4syst-ex:connectsSystemInSNThrough has English-tagged label "connects system in SN through".@en.

A sub-property of the s4syst:connectsSystemThrough property shall have an English-tagged comment (rdfs:comment) that describes in natural language how the connected s4syst:System interact.

EXAMPLE 2: s4syst-ex:connectsSystemInSNThrough has English-tagged comment "Links a single phase connection point to a three-phase power bus connection with which it is connected with a SN configuration".@en A sub-property of the s4syst:connectsSystemThrough property may have an inverse property. If defined, the English-tagged labels and comments of these two properties shall reflect this aspect.

EXAMPLE 3: s4syst-ex:connectsSystemInSNThrough and s4syst-ex:connectsSystemInSNAt are inverse properties.

A sub-property of the s4syst:connectsSystemThrough property may be defined as disjoint from other sub-properties of s4syst:connectsSystemThrough.

EXAMPLE 4: The following properties are disjoint: s4syst-ex:connectsSystemInStarThrough and s4syst-ex:connectsSystemInTriangleThrough.

6.3.4.5 Sub-properties of s4syst:connectsSystemAt

A specialization of the SAREF4SYST pattern may define a sub-property of the s4syst:connectsSystemAt property. If defined, it shall have an English-tagged label (rdfs:label), and the local name of its IRI shall be a mixed case form of its English-tagged label. The label of the sub-property should use the same morpho-syntactic structure as its super-property.

EXAMPLE 1: s4syst-ex:connectsSystemInStarAt has English-tagged label "connects system in star at".@en.

A sub-property of the s4syst:connectsSystemAt property shall have an English-tagged comment (rdfs:comment) that describes in natural language how the connected s4syst:System interact.

EXAMPLE 2: s4syst-ex:connectsSystemInStarAt has English-tagged comment "Links a three-phase power bus connection to one of the three-phase connection points it connects with a triangle configuration".@en.

A sub-class of s4syst:Connection may have an existential or universal restriction on s4syst:connectsSystemAt or one of its sub-properties to a sub-class of s4syst:ConnectionPoint. If so, then the label or the comment of the sub-class of s4syst:Connection and the sub-class of s4syst:ConnectionPoint shall show this relation ostensibly.

EXAMPLE 3: s4syst-ex:USBCommunicationConnection has a universal restriction on s4syst:connectsSystemAt to the class s4syst-ex:USBCommunicationConnectionPoint, and has English-tagged comment "The class of USB communication connections between communication devices".@en.

A sub-property of the s4syst:connectsSystemAt property may have an inverse property. If defined, the English-tagged labels and comments of these two properties shall reflect this aspect.

EXAMPLE 4: `s4syst-ex:connectsSystemInSNAt` and `s4syst-ex:connectsSystemInSNThrough` are inverse properties.

A sub-property of the `s4syst:connectsSystemAt` property may be defined as disjoint from other sub-properties of `s4syst:connectsSystemThrough`.

EXAMPLE 5: The following properties are disjoint: `s4syst-ex:connectsSystemInStarAt` and `s4syst-ex:connectsSystemInTriangleAt.`

6.3.5 Examples for the Smart Grid domain and the Smart Building domain

Different examples of applications of the SAREF4SYST pattern can be found at https://saref.etsi.org/saref4syst/v1.1.2/example/, including for the Smart Grid domain and the Smart Building domain.

The sources of the ontology and the examples can be found at https://saref.etsi.org/sources/saref4syst.

7 Conclusion

The present document specifies 11 SAREF reference ontology patterns in SAREF Core and 3 in SAREF4SYST.

Each sheds light on a specific subset of entities, and provides detailed instructions on how it may or should be be used and specialized in SAREF extensions and SAREF applications. The overall goal is to facilitate the reuse of SAREF Core and ensure modelling homogeneity among the different extensions.

More patterns may be defined in subsequent versions of the present document. These may describe how future entities in SAREF Core may be used and specialized (new clauses in clause 5). They may also describe additional patterns from other SAREF extensions (new clauses after clause 6).

History

Document history		
V1.1.1	July 2019	Publication
V1.1.2	June 2020	Publication
V1.2.1	January 2024	Publication