ETSI TS 103 410-12 V1.1.1 (2023-11)

SmartM2M;
Extension to SAREF;
Part 12: Smart Grid Domain

Reference
DTS/SmartM2M-103410-12SRF4GRID

Keywords
IIoT, IoT, oneM2M, SAREF, semantic, Smart Grid

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).

The present document is part 12 of a multi-part deliverable covering SmartM2M; Extension to SAREF, as identified below:

Part 1: "Energy Domain";

Part 2: "Environment Domain";

Part 3: "Building Domain";

Part 4: "Smart Cities Domain";

Part 5: "Industry and Manufacturing Domains";

Part 6: "Smart Agriculture and Food Chain Domain";

Part 7: "Automotive Domain";

Part 8: "eHealth/Ageing-well Domain";

Part 9: "Wearables Domain";

Part 10: "Water Domain";

Part 11: "Lift Domain";

Part 12: "Smart Grid Domain".

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 presents SAREF4GRID, an extension of SAREF for the Smart Grid domain.

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 (V3.1.1): "SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping".

[2] ETSI TS 118 112 (V3.7.3): "oneM2M; Base Ontology; (oneM2M TS-0012 version 3.7.3 Release 3)".

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] ETSI TR 103 904 (V1.1.1): "SmartM2M; SAREF extension investigation; Requirements for the Smart Grid domain".

[i.2] IEC 62056-1-0:2014: "Electricity metering data exchange - The DLMS/COSEM suite - Part 1-0: Smart metering standardisation framework".

[i.3] IEC 62056-6-1:2017: "Electricity metering data exchange - The DLMS/COSEM suite - Part 6-1: Object Identification System (OBIS)".

[i.4] IEC 62056-6-2:2017: "Electricity metering data exchange - The DLMS/COSEM suite - Part 6-2: COSEM interface classes".

[i.5] W3C® Recommendation 18 August 2009: "SKOS Simple Knowledge Organization System - Reference", A. Miles and S. Bechhofer.

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 explicit capture the semantics of a certain reality

3.2 Symbols

Void.

3.3 Abbreviations

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

COSEM	Companion Specification for Energy Metering
DLMS	Device Language Message Specification
GPS	Global Positioning System
MAC	Media Access Control
OBIS	OBject Identification System
OLTC	On-Load Tap Changer
OWL	Web Ontology Language
PRIME	PoweRline Intelligent Metering Evolution
RDF	Resource Description Framework
RDF-S	Resource Description Framework Schema
SAREF	Smart Applications REFerence ontology
SAREF4GRID	SAREF extension for the Smart Grid domain
SKOS	Simple Knowledge Organization System
UML	Unified Modelling Language
UTC	Universal Time Coordinated

4 SAREF4GRID ontology and semantics

4.1 Introduction and overview

The present document is a technical specification of SAREF4GRID, an extension of SAREF for the Smart Grid domain. This extension has been created by investigating resources from potential stakeholders of the ontology, such as standardization initiatives, associations, and existing ontologies and standards, as reported in ETSI TR 103 904 [i.1]. In addition, the use cases defined in ETSI TR 103 904 [i.1] were also considered, namely:

- **Use case 1:** Remote management of meters.
- **Use case 2:** Management of tertiary sensor devices.
- **Use case 3:** Management of OLTC transformers.
- **Use case 4:** Detection of meter connectivity.

SAREF4GRID is an OWL ontology that extends SAREF and reuses two other ontologies. SAREF4GRID includes 56 classes (41 defined in SAREF4GRID and 15 reused from the SAREF and oneM2M), 50 object properties (28 defined in SAREF4GRID and 22 reused SAREF from and oneM2M), 45 data type properties (43 defined in SAREF4GRID and 2 reused from SAREF and oneM2M), and 28 individuals.

SAREF4GRID focuses on extending SAREF in order to create a common core of general concepts for smart grid data oriented to the IoT field. The main idea is to identify the core components, as mentioned, that could be extended for particular smart grid subdomains, for example, for high voltage networks.

The prefixes and namespaces used in SAREF4GRID and in the present document are listed in Table 1.

Table 1: Prefixes and namespaces used within the SAREF4GRID ontology

Prefix	Namespace
s4grid	https://saref.etsi.org/saref4grid/
saref	https://saref.etsi.org/core/
om	http://www.ontology-of-units-of-measure.org/resource/om-2/
oneM2M	https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl/
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
skos	http://www.w3.org/2008/05/skos#
xsd	http://www.w3.org/2001/XMLSchema#

4.2 SAREF4GRID

4.2.1 General Overview

An overview of the SAREF4GRID ontology is provided in Figures 1, 2, 3 and 4. For all the entities described in the present document, it is indicated whether they are defined in the SAREF4GRID extension or elsewhere by the prefix included before their identifier, i.e. if the element is defined in SAREF4GRID, the prefix is *s4grid*, while if the element is reused from another ontology it is indicated by a prefix according to Table 1.

Arrows are used to represent properties between classes and to represent some RDF, RDF-S and OWL constructs, more precisely:

- Plain arrows with white triangles represent the *rdfs:subClassOf* relation between two classes. The origin of the arrow is the class to be declared as subclass of the class at the destination of the arrow.
- Dashed arrows between two classes indicate a local restriction in the origin class, i.e. that the object property can be instantiated between the classes in the origin and the destination of the arrow. The identifier of the object property is indicated within the arrow.
- Dashed arrows with no identifier are used to represent the *rdf:type* relation, indicating that the element in the origin of the arrow is an instance of the class in the destination of the arrow.

Datatype properties are denoted by rectangles attached to the classes, in an UML-oriented way. Dashed boxes represent local restrictions in the class, i.e. datatype properties that can be applied to the class they are attached to.

Individuals are denoted by rectangles in which the identifier is underlined.

Note that Figures 1, 2, 3 and 4 aim at showing a global overview of the main classes of SAREF4GRID and their mutual relations. More details on the different parts of the figures are provided from clause 4.2.2 to clause 4.2.14.

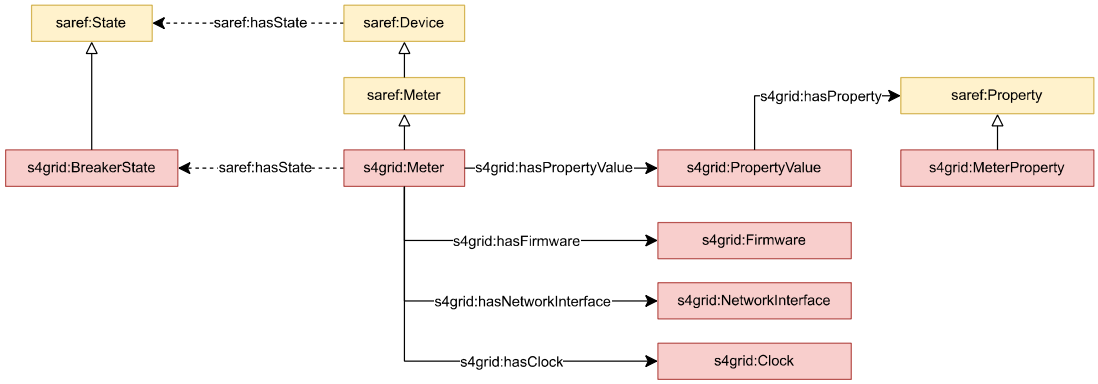

Figure 1: SAREF4GRID overview: Meter information

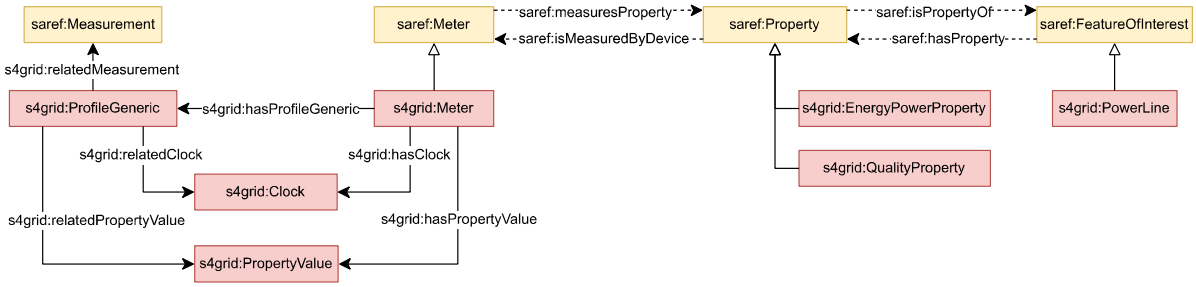

Figure 2: SAREF4GRID overview: Measurements and profiles

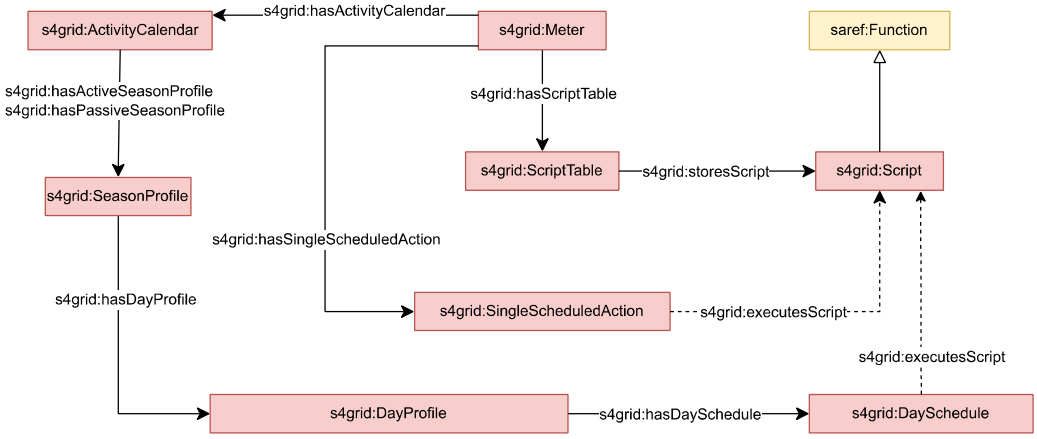

Figure 3: SAREF4GRID overview: Activity calendar and scripts

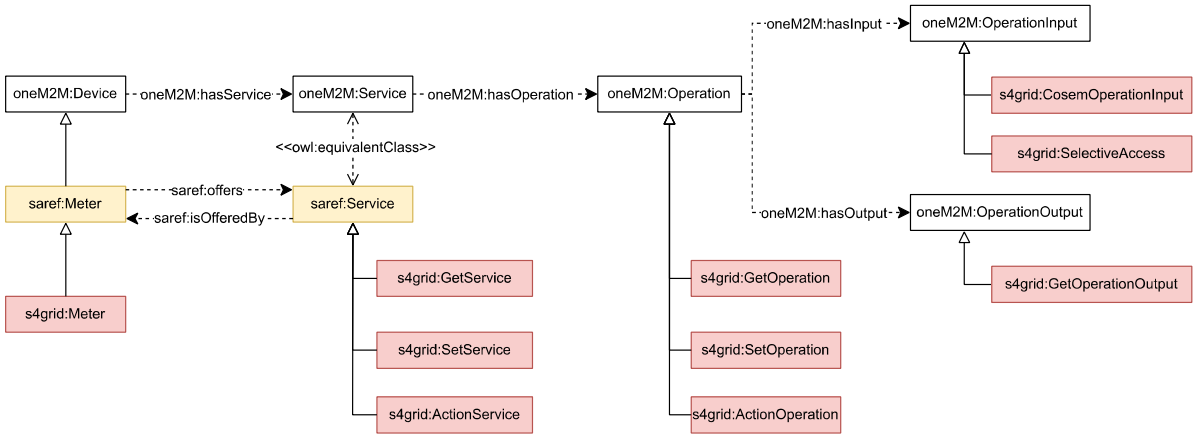

Figure 4: SAREF4GRID overview: Services

4.2.2 Meter

Figure 5 provides an overview of how to represent an electric grid meter using the *s4grid:Meter* class. The representation of electric grid meters and their properties has been extracted from the DLMS/COSEM standard (IEC 62056-1-0:2014 [i.2]).

Unlike in other SAREF extensions, meter-specific information is not defined using properties from SAREF. This is because the DLMS/COSEM standard defines the data structures to model meters from simple up to very complex functionality (IEC 62056-6-2:2017 [i.4]). Moreover, each piece of information within the metering equipment has a unique identifier called OBIS (OBject Identification System) which identifies the instance of a COSEM object (IEC 62056-6-1:2017 [i.3]). This data includes not only measurement values, but also abstract values used for configuration or for obtaining information about the behaviour of the metering equipment.

For this reason, the characteristics of the meter are represented as properties that are not observable by the meter (*s4grid:MeterProperty*, fully represented in Figure 6), i.e. they are not measurements (*saref:Measurement)*. The properties of a meter are defined by a value (*s4grid:PropertyValue*) and some are complemented with a unit of measurement (*saref:UnitOfMeasure*).

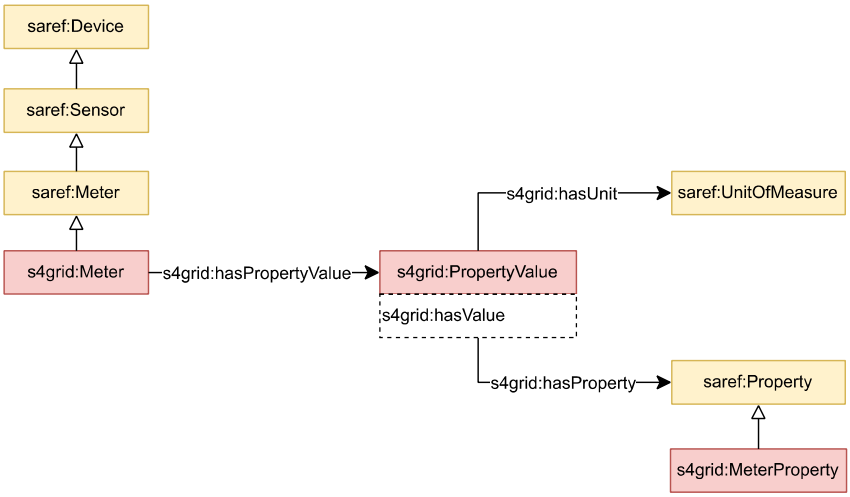

Figure 5: Meter model

Table 2 describes one of the properties that characterize the *s4grid:Meter* class.

Table 2: Property of the s4grid:Meter class related to property value

Property	Definition
s4grid:hasPropertyValue	The value (which may have a unit of measurement) that is related to a property which is not observable by the meter (i.e. is not a measurement).

Table 3 summarizes the properties that characterize the *s4grid:PropertyValue* class.

Table 3: Properties of the s4grid:PropertyValue class

Property	Definition
s4grid:hasProperty	A relationship between a property value and the property it relates to.
s4grid:hasUnit	A relationship identifying the unit of measure used for a certain property value.
s4grid:hasValue	A relationship defining the value of a certain property which is not observable by the meter.

Meters store internal configuration parameters. The DLMS/COSEM standard (IEC 62056-1-0:2014 [i.2]) defines properties related to the configuration of a meter that are necessary to ensure its correct operation. SAREF4GRID categorizes the main properties related to the configuration of a meter (*s4grid:MeterProperties*): screen display configuration (*s4grid:ScreenDisplay*), electric threshold values (*s4grid:Threshold*), time from which a measure has to be outside the threshold before to be considered a quality issue (*s4grid:TimeThreshold*), number of voltage sags (*s4grid:VoltageSagNumber*), number of voltage swells (*s4grid:VoltageSwellNumber*), number of long power failures (*s4grid:LongPowerFailuresNumber*), information provided by the manufacturer (*s4grid:Manufacturer*), turn ratio of the transformer (*s4grid:TransformerRatio*), communication configuration (*s4grid:Network*), status of meter profiles (*s4grid:ProfileStatus*), client power limits (*s4grid:PowerLimit*), reference values for power quality (*s4grid:PowerQuality*), client billing periods (*s4grid:BillingPeriod*), information about the electric grid phase (*s4grid:Phase*), information about the electric grid phase angle (*s4grid:PhaseAngle*), and electric quadrant (*s4grid:Quadrant*). It should be noted that in SAREF4GRID only the general properties are being defined. In order to use a more specific property it is advisable to indicate the general property it comes from (if it exists). The properties which are defined in SAREF4GRID are depicted in Figure 6.

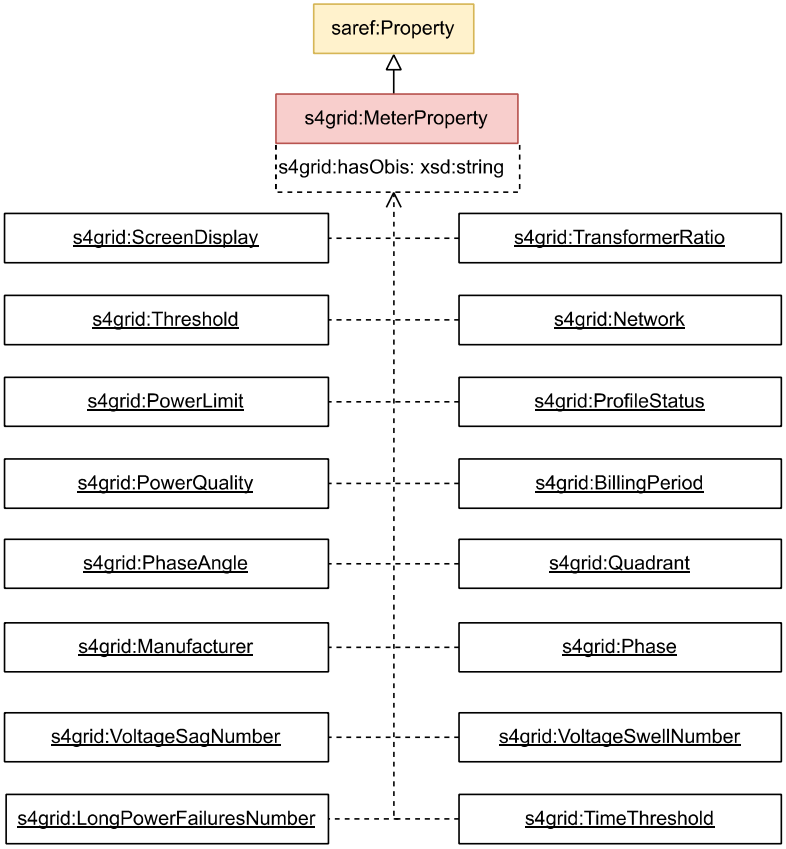

Figure 6: Meter property model

Table 4 summarizes the properties that characterize the *s4grid:MeterProperty* class.

Table 4: Property of the s4grid:MeterProperty class

Property	Definition
s4grid:hasObis	It identifies the instance (COSEM object) of a class. The value of this property conforms to OBIS. OBject Identification System (OBIS) provides a unique identifier for all data within the metering equipment, including not only measurement values, but also abstract values used for configuration or obtaining information about the behaviour of the metering equipment.

4.2.3 Firmware

SAREF4GRID allows describing the identification information related to administration and maintenance of meters by means of the *s4grid:Firmware* class, as presented in Figure 7. They are not communication parameters but support device management. The representation of the firmware of a meter has been extracted from the IEC 62056-6-2:2017 [i.4].

A meter firmware may be described by its: version (*s4grid:hasFirmwareVersion*), unique vendor identifier (*s4grid:hasVendorId*), and unique product identifier as assigned by the vendor (*s4grid:hasProductId*). Besides, a firmware can be related to an electric grid meter by means of the *s4grid:hasFirmware* property.

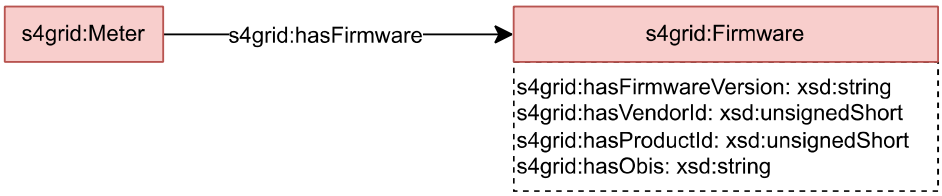

Figure 7: Firmware model

Table 5 describes one of the properties that characterize the *s4grid:Meter* class.

Table 5: Property of the s4grid:Meter class related to firmware

Property	Definition
s4grid:hasFirmware	Holds identification information related to administration and maintenance of meters. They are not communication parameters but allow the device management.

Table 6 summarizes the properties that characterize the *s4grid:Firmware* class.

Table 6: Properties of the s4grid:Firmware class

Property	Definition
s4grid:hasFirmwareVersion exactly 1	Textual description of the firmware version running on the device.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.
s4grid:hasProductId exactly 1	Vendor-assigned unique identifier for the specific product.
s4grid:hasVendorId exactly 1	Unique vendor identifier assigned by the PRIME Alliance.

4.2.4 Network interface

SAREF4GRID allows describing the MAC address of the physical device (or, more generally, of a device or software) by means of the *s4grid:NetworkInterface* class, as presented in Figure 8. There shall be an instance of this class for each network interface of a meter. The representation of the network interface of a meter has been extracted from the IEC 62056-6-2:2017 [i.4].

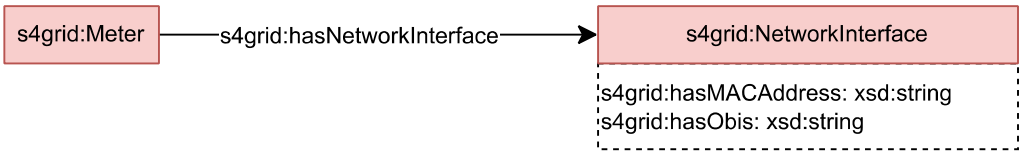

Figure 8: Network interface model

Table 7 describes one of the properties that characterize the *s4grid:Meter* class.

Table 7: Property of the s4grid:Meter class related to network interface

Property	Definition
s4grid:hasNetworkInterface	This class holds the MAC address of the physical device (or, more generally, of a device or software).

Table 8 summarizes the properties that characterize the *s4grid:NetworkInterface* class.

Table 8: Properties of the s4grid:NetworkInterface class

Property	Definition
s4grid:hasMACAddress exactly 1	Holds the MAC address.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.

4.2.5 Clock

SAREF4GRID allows describing the clock of a meter by means of the *s4grid:Clock* class, as presented in Figure 9. This clock manages all information related to date and time including deviations of the local time to a generalized time reference (UTC) due to time zones and daylight-saving time schemes. The representation of the meter clock has been extracted from the IEC 62056-6-2:2017 [i.4].

A meter clock may be described by its: time (*s4grid:hasTime*), time zone where the meter is located (*s4grid:hasTimeZone*), clock status maintained by the meter (*s4grid:hasStatus*), date at which the local time starts to deviate from the normal time (*s4grid:hasDaylightSavingsBegin*), date at which the local time ends to deviate from the normal time (*s4grid:hasDaylightSavingsEnd*), deviation in generalized time (*s4grid:hasDaylightSavingsDeviation*), if the daylight savings time feature is enabled (*s4grid:hasDaylightSavingsEnabled*), and where the basic timing information comes from (*s4grid:hasClockBase*). Besides, a clock can be related to a meter by means of the *s4grid:hasClock* property.

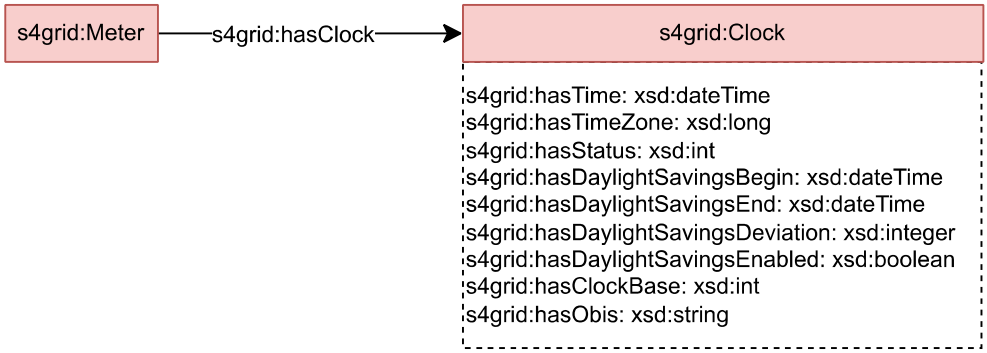

Figure 9: Clock model

Table 9 describes one of the properties that characterize the *s4grid:Meter* class.

Table 9: Property of the s4grid:Meter class related to clock

Property	Definition
s4grid:hasClock	This class models the device clock, managing all information related to date and time including deviations of the local time to a generalized time reference (UTC) due to time zones and daylight-saving time schemes.

Table 10 summarizes the properties that characterize the *s4grid:Clock* class.

Table 10: Properties of the s4grid:Clock class

Property	Definition
s4grid:hasClockBase exactly 1	Defines where the basic timing information comes from: (0) not defined; (1) internal crystal; (2) mains frequency 50 Hz; (3) mains frequency 60 Hz; (4) GPS (global positioning system); (5) radio controlled.
s4grid:hasDaylightSavingsBegin exactly 1	Defines the local switch date and time when the local time starts to deviate from the normal time.
s4grid:hasDaylightSavingsDeviation exactly 1	Contains the number of minutes by which the deviation in generalized time shall be corrected at daylight savings begin.
s4grid:hasDaylightSavingsEnabled exactly 1	Enable and disable the daylight savings time feature.
s4grid:hasDaylightSavingsEnd exactly 1	Defines the local switch date and time when the local time ends to deviate from the local normal time.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.
s4grid:hasStatus exactly 1	Status of the clock.
s4grid:hasTime exactly 1	Meter's local date and time.
s4grid:hasTimeZone exactly 1	The deviation of local, normal time to UTC in minutes. The value depends on the geographical location of the meter.

4.2.6 Breaker state

As it can be observed in Figure 10, the modelling of states in the SAREF4GRID ontology mostly relies on the state model proposed in SAREF. In order to reduce duplication with SAREF documentation, the reader is referred to the SAREF specification ETSI TS 103 264 [1] for details about state modelling including here details only for the new concepts.

SAREF allows to define the state in which a device can be found. However, the SAREF4GRID extension also requires to be able to define the possible transitions between states and complex states. Therefore, the *s4grid:BreakerState* class has been defined according to the IEC 62056-6-2:2017 [i.4].

A meter breaker represents the internal or external disconnect unit of the meter (e.g. electricity breaker, gas valve) in order to connect or disconnect the premises of the consumer to/from the supply. A meter breaker state may be described by its: physical state (*s4grid:hasOutputState*), internal state (*s4grid:hasControlState*) and the possible transitions between states (*s4grid:hasControlMode*). For more information between the possible transitions see the IEC 62056-6-2:2017 [i.4].

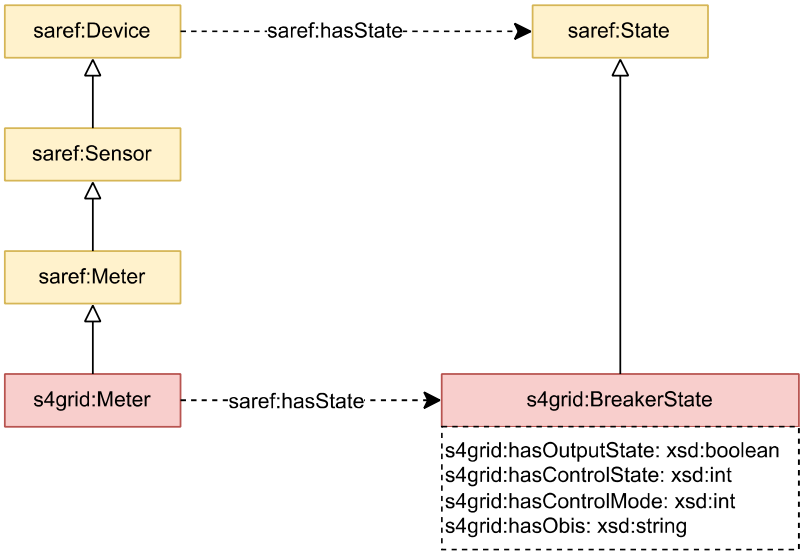

Figure 10: Breaker state model

Table 11 summarizes the properties that characterize the *s4grid:BreakerState* class.

Table 11: Properties of the s4grid:BreakerState class

Property	Definition
s4grid:hasControlMode exactly 1	Configures the behaviour of the disconnect control object for all triggers, i.e. the possible state transitions.
s4grid:hasControlState exactly 1	Shows the internal state of the disconnect control object: (0) Disconnected; (1) Connected; (2) Ready for reconnection.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.
s4grid:hasOutputState exactly 1	Shows the actual physical state of the device connection the supply: (True) Connected, (False) Disconnected.

4.2.7 Script table

As it can be observed in Figure 11, the modelling of scripts in the SAREF4GRID ontology mostly relies on the service model proposed in SAREF. In order to reduce duplication with SAREF documentation, the reader is referred to the SAREF specification [1] for details about service modelling including here details only for the new concepts.

SAREF allows to define the functions which accomplish the task for which a device is designed. However, the SAREF4GRID extension also requires to be able to define the triggering of a series of actions by executing scripts, and where those scripts are stored. Therefore, the *s4grid:ScriptTable* class has been defined according to the IEC 62056-6-2:2017 [i.4].

A script table represents a table of script entries. Moreover, the *s4grid:Script* class defines a series of action specifications. An action specification activates a method or modifies an attribute of a COSEM object within the logical device. Besides, a script table can be related to an electric grid meter by means of the *s4grid:hasScriptTable* property.

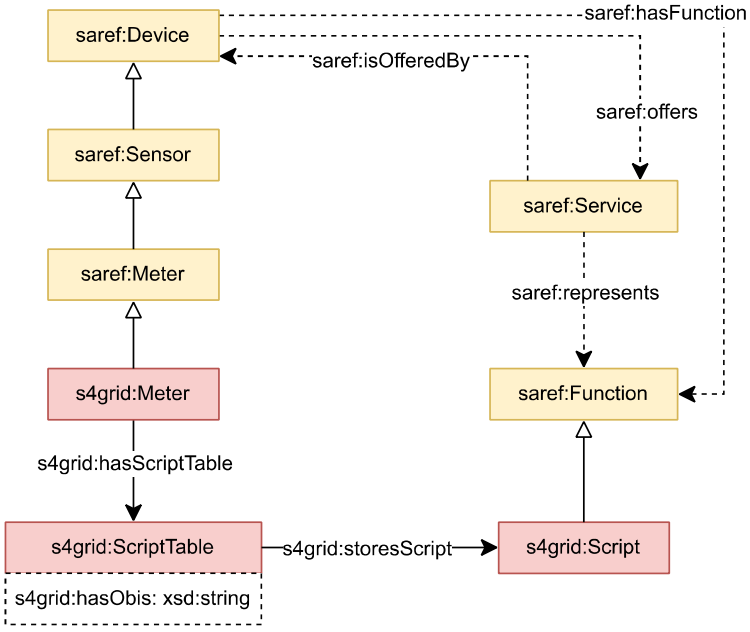

Figure 11: Script table model

Table 12 describes one of the properties that characterize the *s4grid:Meter* class.

Table 12: Property of the s4grid:Meter class related to script table

Property	Definition
s4grid:hasScriptTable	The triggering of a series of actions by executing scripts.

Table 13 summarizes the properties that characterize the *s4grid:ScriptTable* class.

Table 13: Properties of the s4grid:ScriptTable class

Property	Definition
s4grid:storesScript	Specifies the different scripts which are stored in a script table.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.

4.2.8 Scheduled action

SAREF4GRID allows the execution of periodic actions within a meter by means of the *s4grid:SingleScheduledAction* class, as presented in Figure 12; such actions are not necessarily linked to tariffication. A scheduled action describes the script, which is stored in a script table, that is going to be executed at a determined date. The representation of the meter scheduled action has been extracted from the IEC 62056-6-2:2017 [i.4].

A meter single scheduled action may be described by its: execution time (*s4grid:hasExecutionTime*) and what script is going to be executed (*s4grid:executesScript*). Besides, a single schedule action can be related to a meter by means of the *s4grid:hasSingleScheduledAction* property.

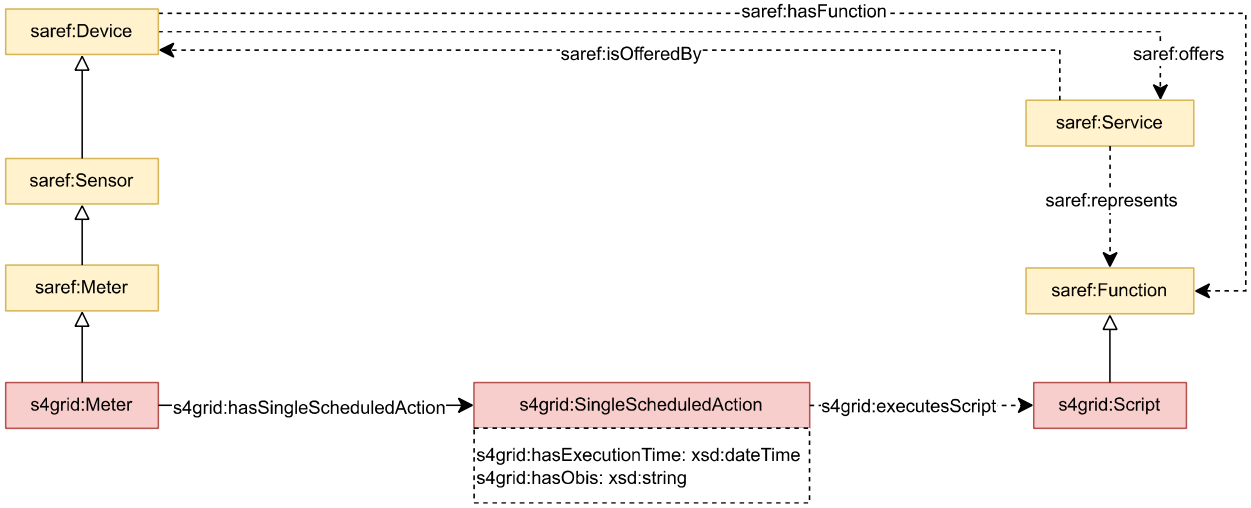

Figure 12: Scheduled action model

Table 14 describes one of the properties that characterize the *s4grid:Meter* class.

Table 14: Property of the s4grid:Meter class related to scheduled action

Property	Definition
s4grid:hasSingleScheduledAction	The execution of periodic actions within a meter; such actions are not necessarily linked to tariffication.

Table 15 summarizes the properties that characterize the *s4grid:SingleScheduledAction* class.

Table 15: Properties of the s4grid:SingleScheduledAction class

Property	Definition
s4grid:executesScript exactly 1	Defines the script to be executed.
s4grid:hasExecutionTime	Specifies the time and the date when the script is executed.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.

4.2.9 Activity calendar

SAREF4GRID allows modelling the handling of various tariff structures in the meter by means of the *s4grid:ActivityCalendar* class, as presented in Figure 13. An activity calendar provides a list of scheduled actions, following the classical way of calendar-based schedules by defining seasons, weeks, etc. The representation of the meter activity calendar has been extracted from the IEC 62056-6-2:2017 [i.4].

An activity calendar is active (*s4grid:hasCalendarNameActive*) if it is currently used for billing. Each active calendar has an associated passive calendar (*s4grid:hasCalendarNamePassive*) and its function is to allow to modify the parameters of the active calendar on a date prior to its activation date (*s4grid:hasActivatePassiveCalendarTime).* Activation date is the date from which the meter will use the passive calendar parameters and, therefore, they become active calendar parameters. An active calendar is compound by active seasons (*s4grid:hasActiveSeasonProfile*) and a passive calendar is compound by passive seasons (*s4grid:hasPassiveSeasonProfile*). Notice that there is no distinction between an active calendar and a passive calendar because together they represent an activity calendar and they share the same OBIS code.

A season profile (*s4grid:SeasonProfile*) represents periods of time during the year when billing conditions are always the same. A season profile is characterized by a start date (*s4grid:hasSeasonStart*) and seven day profiles (*s4grid:hasDayProfile*) to apply, which together represent a week (there is one day profile for each day of the week). A season profile finishes when the next season profile begins.

A day profile (*s4grid:DayProfile)* represents the discrimination of time along the day. Moreover, the seven day profiles together represent a period during the week when billing conditions are always the same. There are two day profiles: regular days (*s4grid:RegularDayProfile*), which represent not festive days, and special days (*s4grid:SpecialDayProfile*), which represent at which date there is a festivity, i.e. normal day behaves as a special day (*s4grid:hasScpecialDayDate)*. A day profile is characterized by a day schedule (*s4grid:hasDaySchedule).*

A day schedule *(s4grid:DaySchedule)* defines the activation of certain scripts during the day, which can perform different activities inside the meter. For each day schedule, a list of scheduled actions is defined by a script to be executed (*s4grid:executesScript)* with the corresponding activation time (*s4grid:hasStartTime).*

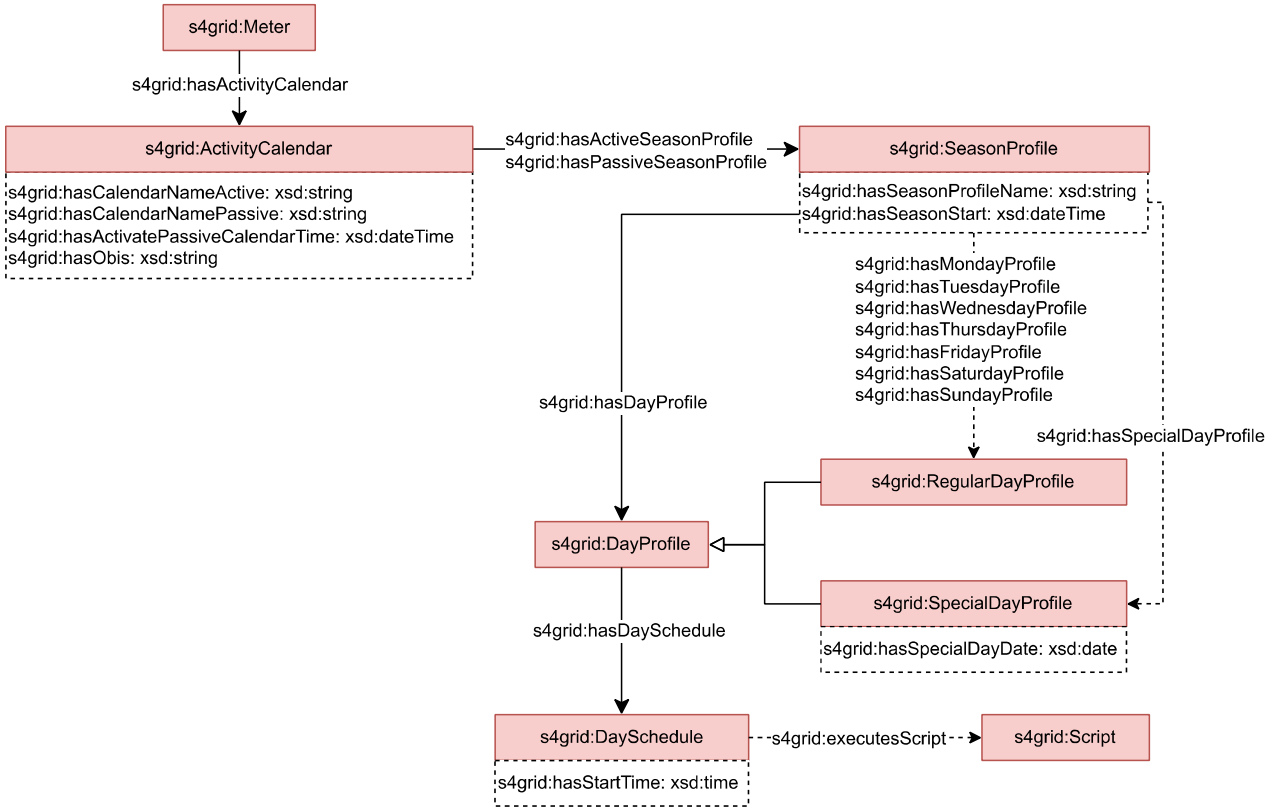

Figure 13: Activity calendar model

Table 16 describes one of the properties that characterize the *s4grid:Meter* class.

Table 16: Property of the s4grid:Meter class related to activity calendar

Property	Definition
s4grid:hasActivityCalendar	Allows handling various tariff structures in the meter. This class provides a list of scheduled actions, following the classical way of calendar-based schedules by defining seasons, weeks, etc.

Table 17 summarizes the properties that characterize the *s4grid:ActivityCalendar* class.

Table 17: Properties of the s4grid:ActivityCalendar class

Property	Definition
s4grid:hasActiveSeasonProfile	Contains a list of active season profiles. These season profiles form an active calendar.
s4grid:hasPassiveSeasonProfile	Contains a list of passive season profiles. These season profiles form a passive calendar.
s4grid:hasActivatePassiveCalendarTime max 1	Defines the time when the passive calendar will be activated.
s4grid:hasCalendarNameActive exactly 1	User defined name identifying the passive calendar. Typically contains an identifier of the active calendar (currently active), which is descriptive to the set of scripts activated by the object.
s4grid:hasCalendarNamePassive exactly 1	User defined name identifying the passive calendar. Typically contains an identifier of the passive calendar (will be activated), which is descriptive to the set of scripts activated by the object.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.

Table 18 summarizes the properties that characterize the *s4grid:SeasonProfile* class.

Table 18: Properties of the s4grid:SeasonProfile class

Property	Definition
s4grid:hasDayProfile	List of scheduled actions which are defined by a script to be executed and the corresponding activation time.
s4grid:hasSeasonStart exactly 1	Starting time of the season profile. The current season profile is terminated by the starting time of the next season profile.
s4grid:hasSeasonProfileName exactly 1	User defined name identifying the current season profile.
s4grid:hasSpecialDayProfile	It defines when there is a festivity and therefore a normal day behaves as a special date.
s4grid:hasMondayProfile exactly 1	It defines the day profile valid each Monday of the season.
s4grid:hasTuesdayProfile exactly 1	It defines the day profile valid each Tuesday of the season.
s4grid:hasWednesdayProfile exactly 1	It defines the day profile valid each Wednesday of the season.
s4grid:hasThursdayProfile exactly 1	It defines the day profile valid each Thursday of the season.
s4grid:hasFridayProfile exactly 1	It defines the day profile valid each Friday of the season.
s4grid:hasSaturdayProfile exactly 1	It defines the day profile valid each Saturday of the season.
s4grid:hasSundayProfile exactly 1	It defines the day profile valid each Sunday of the season.

Table 19 summarizes the properties that characterize the *s4grid:SpecialDayProfile* class.

Table 19: Property of the s4grid:SpecialDayProfile class

Property	Definition
s4grid:hasSpecialDayDate exactly 1	Dates at which a normal day is considered a special date (i.e. Christmas).

Table 20 summarizes the properties that characterize the *s4grid:DayProfile* class.

Table 20: Property of the s4grid:DayProfile class

Property	Definition
s4grid:hasDaySchedule	Scheduled actions to be executed at a given time for a day profile.

Table 21 summarizes the properties that characterize the *s4grid:DaySchedule* class.

Table 21: Properties of the s4grid:DaySchedule class

Property	Definition
s4grid:executesScript exactly 1	Defines the script to be executed.
s4grid:hasStartTime exactly 1	Time when the script is going to be executed.

4.2.10 Power line properties

As it can be observed in Figure 14 and Figure 15, the modelling of measurements in the SAREF4GRID ontology mostly relies on the measurement model proposed in SAREF. In order to reduce duplication with SAREF documentation, the reader is referred to the SAREF specification [1] for details about measurement modelling including here details only for the new concepts.

The DLMS/COSEM standard (IEC 62056-1-0 [i.2]) defines the measurements that a meter shall take from a power line. It should be noted that in SAREF4GRID only the general properties are being defined. In order to use a more specific property it is necessary to indicate the general property from which it comes from. The properties that are defined in SAREF4GRID, which are measured from a power line (*s4grid:PowerLine*), are depicted in Figure 14 and Figure 15.

SAREF4GRID categorizes the main properties related to the energy and power measurements of a power line (*s4grid:EnergyPowerProperty*): active energy measurements (*s4grid:ActiveEnergy*), reactive energy measurements (*s4grid:ReactiveEnergy*), apparent power measurements (*s4grid:ApparentPower*), demand register measurements (*s4grid:DemandRegister*), active power measurements (*s4grid:ActivePower*), reactive power measurements (*s4grid:ReactivePower*), current measurements (*s4grid:Current*), voltage measurements (*s4grid:Voltage*), and power factor related measurements (*s4grid:PowerFactor*).

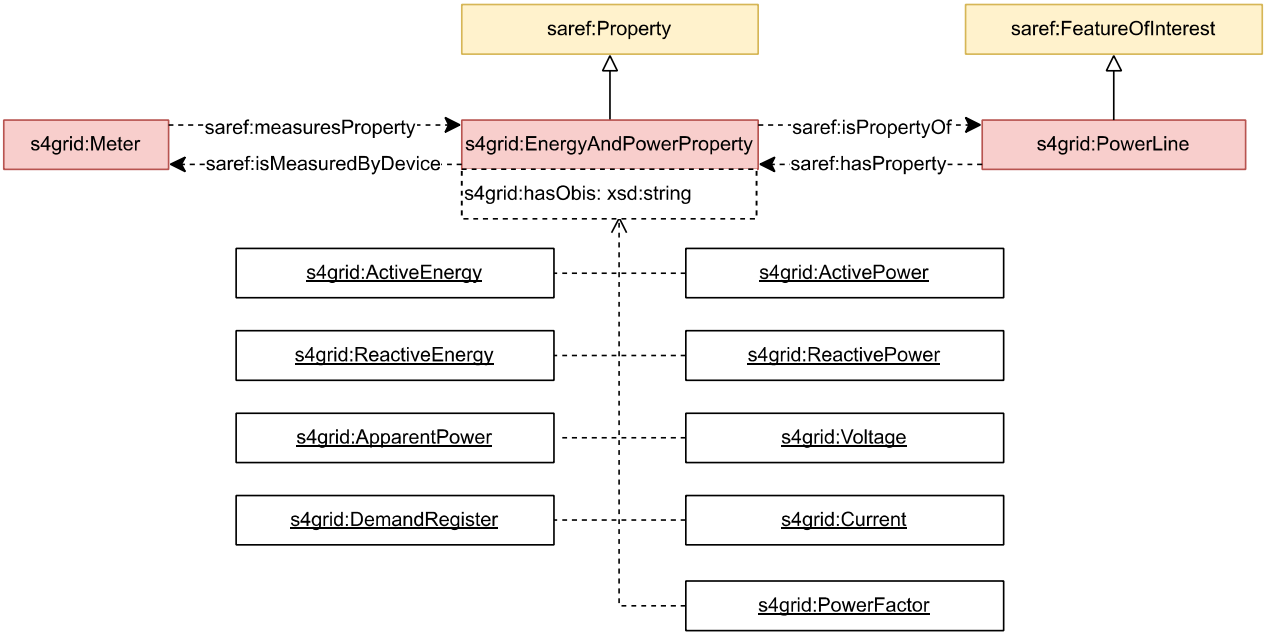

Figure 14: Energy and power property model

Table 22 summarizes the properties that characterize the *s4grid:EnergyPowerProperty* class.

Table 22: Property of the s4grid:EnergyPowerProperty class

Property	Definition
s4grid:hasObis	It identifies the instance (COSEM object) of a class.

SAREF4GRID also categorises the main properties related to the quality measurements of a power line (*s4grid:QualityProperty*): duration of voltage sags (*s4grid:DurationVoltageSag*), duration of voltage swells (*s4grid:DurationVoltageSwell*), and duration of long power failures (*s4grid:DurationLongPowerFailure*).

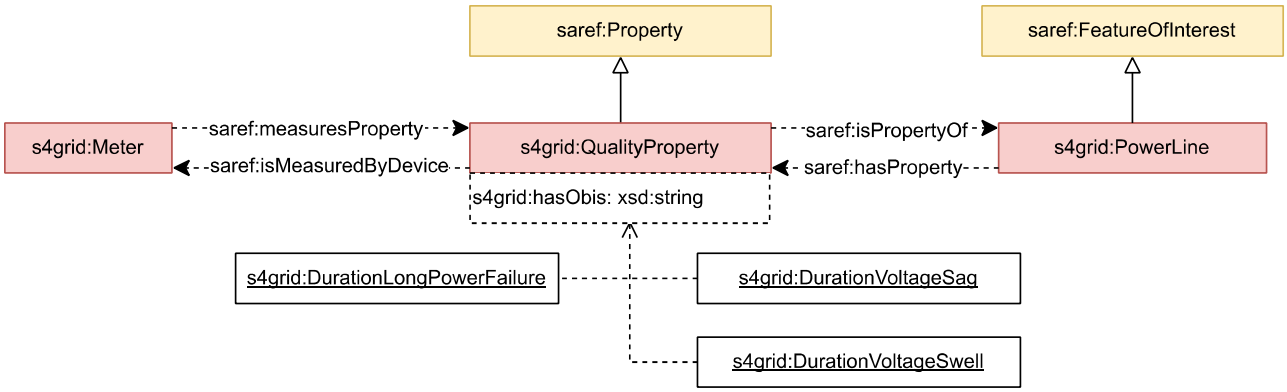

Figure 15: Quality property model

Table 23 summarizes the properties that characterize the *s4grid:QualityProperty* class.

Table 23: Property of the s4grid:QualityProperty class

Property	Definition
s4grid:hasObis	It identifies the instance (COSEM object) of a class.

4.2.11 Profile generic

SAREF4GRID allows modelling the storing, sorting and accessing of data groups or data series (i.e. capture objects in COSEM) in the meter by means of the *s4grid:ProfileGeneric* class, as presented in Figure 16. Capture objects are specific attributes or elements of (an) attribute(s) of COSEM objects. The capture objects are collected periodically or occasionally. The representation of the profile generic of a meter has been extracted from the IEC 62056-6-2:2017 [i.4].

A profile generic is represented by the objects that it captures (*s4grid:Clock*, *s4grid:PropertyValues* and *s4grid:Measurements*). These capture objects are collected in each period defined in the *s4grid:hasCapturePeriod* property.

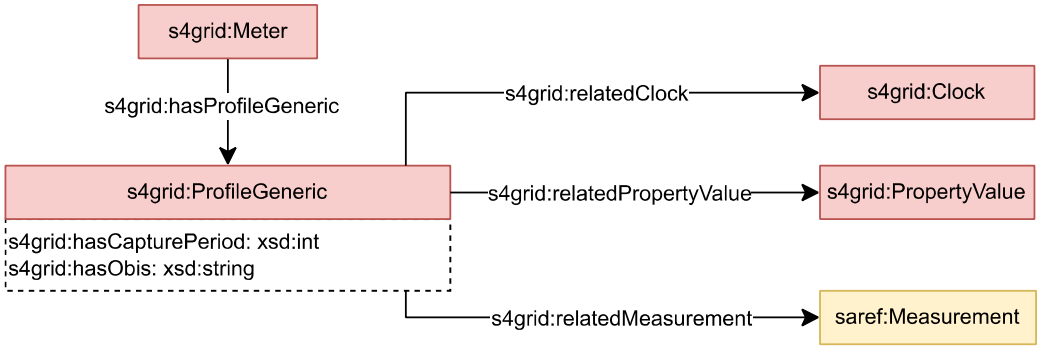

Figure 16: Profile generic model

Table 24 describes one of the properties that characterize the *s4grid:Meter* class.

Table 24: Property of the s4grid:Meter class related to profile generic

Property	Definition
s4grid:hasProfileGeneric	The class that provides a generalized concept allowing to store, sort and access data groups or data series, called capture objects. Capture objects are appropriate attributes or elements of (an) attribute(s) of COSEM objects. The capture objects are collected periodically or occasionally.

Table 25 summarizes the properties that characterize the *s4grid:ProfileGeneric* class.

Table 25: Property of the s4grid:ProfileGeneric class

Property	Definition
s4grid:relatedClock	The clock that is captured by a profile generic.
s4grid:relatedMeasurement	The measurement that is captured by a profile generic.
s4grid:relatedPropertyValue	The property value that is captured by a profile generic.
s4grid:hasCapturePeriod exactly 1	The capturing period of a profile generic in seconds. If its value is 0, then there is no automatic capturing; capturing is triggered externally or through capture events that occur asynchronously. If its value is greater than 0, then automatic capturing is assumed.
s4grid:hasObis	It identifies the instance (COSEM object) of a class.

4.2.12 Get service

Figure 17 provides an overview of the modelling of get services (*s4grid:GetService*). A get service is performed through get operations (*s4grid:GetOperation*). The get operation modelling involves two main concepts, namely *s4grid:CosemOperationInput* and *s4grid:GetOperationOutput*. As can be seen in the figure, the modelling of get services totally relies on the service model proposed in ETSI TS 118 112 [2]. In order to reduce duplication with the oneM2M documentation, the reader is referred to the oneM2M specification for details about service modelling. The representation of the inputs and outputs of a get service has been extracted from the IEC 62056-6-2:2017 [i.4].

A get operation needs one input which represents what is going to be retrieved (the whole instance or just a property of the instance). Therefore, the input of a get operation can be either a class, the range of a datatype property, or the range of an object property. The *s4grid:CosemOperationInput* class specifies the instance from which data is going to be retrieved by indicating the OBIS code (*s4grid:obtainInputFromObis*). If only the OBIS code is specified, it is understood that the whole instance is going to be retrieved. Moreover, the *s4grid:GetOperationPropertyInput* class specifies the object/datatype property of an instance from which data is going to be retrieved by indicating the OBIS code and the name of the object/datatype property (*s4grid:obtaintInputForProperty*). If the OBIS code and the property name are specified, it is understood that just the range of a property of the instance is going to be retrieved.

A get operation is going to generate one output which represents the type of what is going to be retrieved. The output of a get operation (*s4grid:GetOperationOutput*) can be either a class or a datatype. The *s4grid:GetOperationDataOutput* class specifies that the output is going to be a datatype. In this case the type of the output is defined using the *s4grid:hasOutputDataType* property, indicating the type of the datatype. The *s4grid:GetOperationObjectOutput* class specifies that the output is going to be a class. In this case the type of the output is defined using the *s4grid:hasOutputObjectType* property, indicating the name of the class.

Additionally, in the case of a get service of a *s4grid:ProfileGeneric* class, a selective access (*s4grid:SelectiveAccess*) can be specified. This indicates the range of entries to be retrieved (*s4grid:EntryDescriptor*) or the range of values (*s4grid:RangeDescriptor)* to be retrieved from a *s4grid:ProfileGeneric* class.

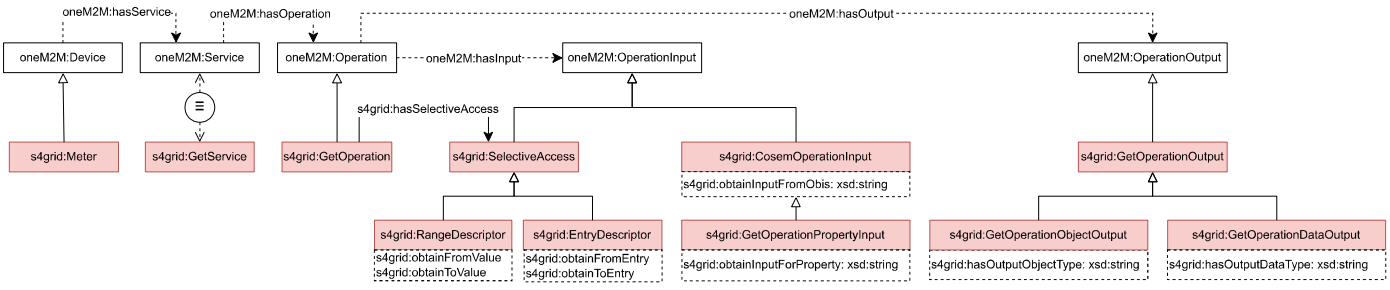

Figure 17: Get service model

Table 26 summarizes the properties that characterize the *s4grid:GetOperation* class.

Table 26: Property of the s4grid:GetOperation class

Property	Definition
s4grid:hasSelectiveAccess	The selective access of a profile generic get service. The get operations property-related services typically reference the entire property. However, in the case of certain properties, selective access to only part of the property may be provided.

Table 27 summarizes the properties that characterize the *s4grid:CosemOperationInput* class.

Table 27: Property of the s4grid:CosemOperationInput class

Property	Definition
s4grid:obtainInputFromObis exactly 1	It identifies the instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to OBIS.

Table 28 summarizes the properties that characterize the *s4grid:GetOperationPropertyInput* class.

Table 28: Property of the s4grid:GetOperationPropertyInput class

Property	Definition
s4grid:obtainInputForProperty exactly 1	It identifies the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to an object property or datatype property of SAREF4GRID.

Table 29 summarizes the properties that characterize the *s4grid:EntryDescriptor* class.

Table 29: Properties of the s4grid:EntryDescriptor class

Property	Definition
s4grid:obtainFromEntry	It defines the first entry to be retrieved from a profile generic.
s4grid:obtainToEntry	It defines the last entry to be retrieved from a profile generic.

Table 30 summarizes the properties that characterize the *s4grid:RangeDescriptor* class.

Table 30: Properties of the s4grid:RangeDescriptor class

Property	Definition
s4grid:obtainFromValue	It restricts the range of entries to be retrieved from a profile generic defining the oldest or smallest entry to retrieve.
s4grid:obtainToValue	It restricts the range of entries to be retrieved from a profile generic defining the newest or largest entry to retrieve.

Table 31 summarizes the properties that characterize the *s4grid:GetOperationDataOutput* class.

Table 31: Property of the s4grid:GetOperationDataOutput class

Property	Definition
s4grid:hasOutputDataType exactly 1	It identifies the type of the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to the range of a datatype property of SAREF4GRID.

Table 32 summarizes the properties that characterize the *s4grid:GetOperationObjectOutput* class.

Table 32: Property of the s4grid:GetOperationObjectOutput class

Property	Definition
s4grid:hasOutputObjectType exactly 1	It identifies the type of the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to the range of an object property of SAREF4GRID.

4.2.13 Set service

Figure 18 provides an overview of the modelling of set services (*s4grid:SetService*). A set service is performed through set operations (*s4grid:SetOperation*). The set operation modelling involves one main concept, namely *s4grid:CosemOperationInput*. As can be seen in the figure, the modelling of set services totally relies on the service model proposed in ETSI TS 118 112 [2]. In order to reduce duplication with the oneM2M documentation, the reader is referred to the oneM2M specification for details about service modelling. The representation of the inputs and outputs of a set service has been extracted from the IEC 62056-6-2:2017 [i.4].

A set operation needs two inputs: the element that is going to be modified and the new data that is going to replace the old data. The element that is going to be modified is represented by a class meanwhile the new data is represented either by a class, the range of a datatype property, or the range of an object property (depending on if the whole instance is going to be modified or just a property). Therefore, the *s4grid:CosemOperationInput* class specifies the instance from which data is going to be modified by indicating the OBIS code (*s4grid:obtaintInputFromObis*) and the new data is represented either by the *s4grid:SetOperationObisInput*, *s4grid:SetOperationObjectInput*, or *s4grid:SetOperationDataInput* classes.

The *s4grid:SetOperationObisInput* class indicates that the whole instance is going to be modified. In this case the type of the input is defined using the *s4grid:hasInputObjectType* property, indicating the name of the class expected to modify the instance. The *s4grid:SetOperationObjectInput* class indicates that just the range of an object property of the instance is going to be modified. In this case the type of the input is defined using the *s4grid:obtainInputForProperty*, which indicates the name of the object property whose range is going to be modified, and the *s4grid:hasInputObjectType* property, indicating the name of the class expected to modify the range of the object property. The *s4grid:SetOperationDataInput* class indicates that just the range of a datatype property of the instance is going to be modified. In this case the type of the input is defined using the *s4grid:obtainInputForProperty*, which indicates the name of the datatype property whose range is going to be modified, and the *s4grid:hasInputDataType* property, which indicates the type of the datatype expected to modify the range of the datatype property.

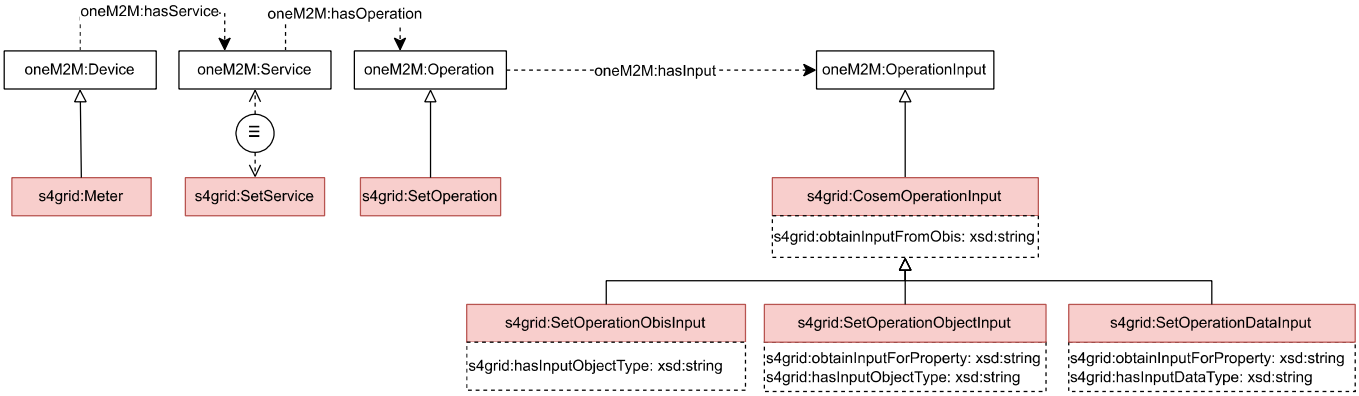

Figure 18: Set service model

Table 33 summarizes the properties that characterize the *s4grid:CosemOperationInput* class.

Table 33: Property of the s4grid:CosemOperationInput class

Property	Definition
s4grid:obtainInputFromObis exactly 1	It identifies the instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to OBIS.

Table 34 summarizes the properties that characterize the *s4grid:SetOperationObisInput* class.

Table 34: Property of the s4grid:SetOperationObisInput class

Property	Definition
s4grid:hasInputObjectType exactly 1	It identifies the type of the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to the range of an object property of SAREF4GRID.

Table 35 summarizes the properties that characterize the *s4grid:SetOperationObjectInput* class.

Table 35: Properties of the s4grid:SetOperationObjectInput class

Property	Definition
s4grid:obtainInputForProperty exactly 1	It identifies the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to and object property or datatype property of SAREF4GRID.
s4grid:hasInputObjectType exactly 1	It identifies the type of the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to the range of an object property of SAREF4GRID.

Table 36 summarizes the properties that characterize the *s4grid:SetOperationDataInput* class.

Table 36: Properties of the s4grid:SetOperationDataInput class

Property	Definition
s4grid:obtainInputForProperty exactly 1	It identifies the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to and object property or datatype property of SAREF4GRID.
s4grid:hasInputDataType exactly 1	It identifies the type of the attribute of an instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to the range of a datatype property of SAREF4GRID.

4.2.14 Action service

Figure 19 provides an overview of the modelling of action services (*s4grid:ActionService*). An action service is performed through action operations (*s4grid:ActionOperation*). The action operation modelling involves one main concept, namely *s4grid:CosemOperationInput*. As can be seen in the figure, the modelling of action services totally relies on the service model proposed in ETSI TS 118 112 [2]. In order to reduce duplication with the oneM2M documentation, the reader is referred to the oneM2M specification for details about service modelling. The representation of the inputs and outputs of an action service has been extracted from the IEC 62056-6-2:2017 [i.4].

An action operation needs two inputs: the element that is going to be affected by an action and the parameters necessary for the action to be executed. The element that is going to be affected by an action is represented by a class and the parameters are either represented by a class or a value. Therefore, the *s4grid:CosemOperationInput* class specifies the instance from which data is going to be modified by indicating the OBIS code (*s4grid:obtainInputFromObis*) and the parameters are represented either by the *s4grid:SimpleActionOperationInput* or the *s4grid:ComplexActionOperationInput* classes. The *s4grid:CosemOperationInput* class specifies the instance that is going to be affected by the action by indicating the OBIS code (*s4grid:obtainInputFromObis*). The *s4grid:SimpleActionOperationInput* class indicates that the parameter needed by the action to operate is simple (i.e. integer, string, etc.). In this case the value of the parameter is defined using the *s4grid:hasActionValue* property. The *s4grid:ComplexActionOperationInput* class indicates that the parameter needed by the action to operate is complex (i.e. structure). There are two cases of complex parameters: *s4grid:PresetAdjustingTime* class, which is needed by a *s4grid:Clock* to modify the time, and *s4grid:SpecialDayEntry* class, which is needed by an *s4grid:ActivityCalendar* to adding a new special day.

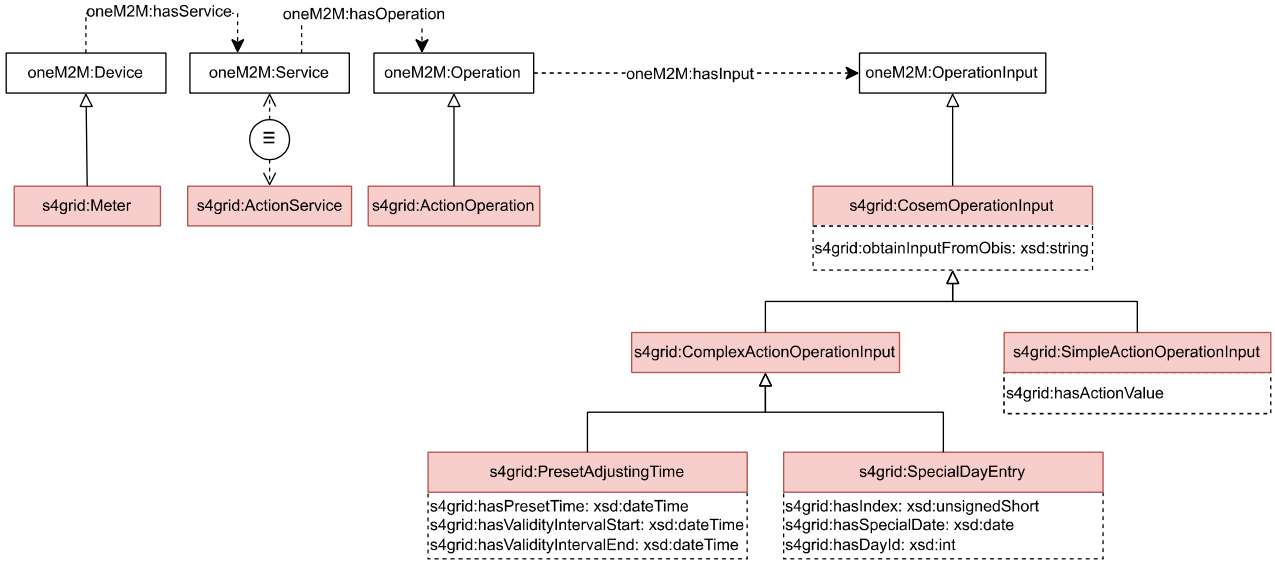

Figure 19: Action service model

Table 37 summarizes the properties that characterize the *s4grid:CosemOperationInput* class.

Table 37: Property of the s4grid:CosemOperationInput class

Property	Definition
s4grid:obtainInputFromObis exactly 1	It identifies the instance (COSEM object) of a class from which the data is going to be obtained. The value of this property conforms to OBIS.

Table 38 summarizes the properties that characterize the *s4grid:SimpleOperationInput* class.

Table 38: Property of the s4grid:SimpleOperationInput class

Property	Definition
s4grid:hasActionValue exactly 1	It specifies the input of a method from the instance (COSEM object) of a class. The value of this property conforms to a simple type (not a structure).

Table 39 summarizes the properties that characterize the *s4grid:PresetAdjustingTime* class.

Table 39: Properties of the s4grid:PresetAdjustingTime class

Property	Definition
s4grid:hasPresetTime exactly 1	It pre-sets the clock time to a new value.
s4grid:hasValidityIntervalEnd exactly 1	It defines the end of a validity interval within which the new time can be activated.
s4grid:hasValidityIntervalStart exactly 1	It defines the start of a validity interval within which the new time can be activated.

Table 40 summarizes the properties that characterize the *s4grid:SpecialDayEntry* class.

Table 40: Properties of the s4grid:SpecialDayEntry class

Property	Definition
s4grid:hasDayId exactly 1	It defines the unique identifier of a day.
s4grid:hasSpecialDate exactly 1	It defines the date at which a normal day is considered a special date.
s4grid:hasIndex exactly 1	It defines the index of the special day.

4.3 Instantiating SAREF4GRID

This clause shows different examples of how to instantiate the SAREF4GRID extension of SAREF.

The example presented in Figure 20 depicts an electric grid meter (*ex:Meter1234*). It can be described by a set of meter properties (such as the one shown in the figure, *ex:ScrollDisplayMode*) that are identified by an OBIS code (*s4grid:hasObis*). Notice that some meter properties do not specify a unit of measure.

SAREF4GRID does not aim to provide an exhaustive definition of all the properties defined in the IEC 62056-6-2 [i.4]. Instead, it defines a set of general properties (those shown in Figure 6, e.g. *s4grid:ScreenDisplay* in the figure) and specific properties can be related to these general properties using the SKOS ontology [i.5]. Using SKOS more specific properties can be defined specifying from which general property they are derived (*skos:narrower*), and which properties belong to a general property (*skos:broader*).

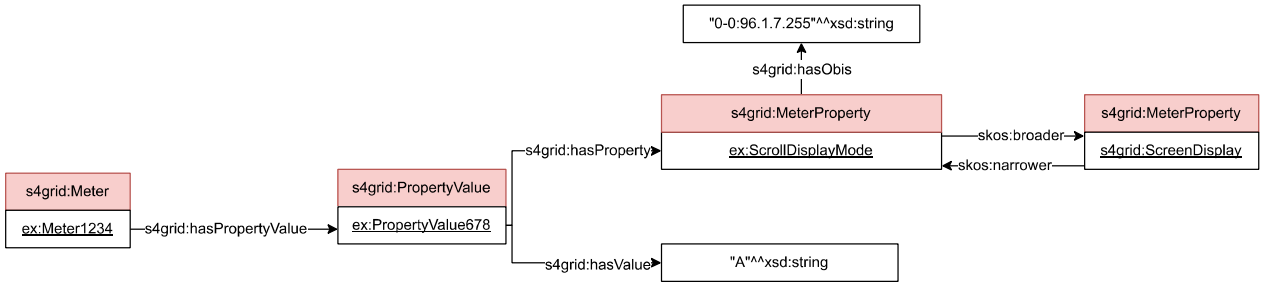

Figure 20: Example of electric grid meter information I

The example presented in Figure 21 depicts an electric grid meter (*ex:Meter1234*). It can be described by a set of meter properties (e.g. *ex:ActivePowerLimitContract1TariffPeriod1*) which are identified by an OBIS code (*s4grid:hasObis*). The meter properties used to describe a meter are broader than the properties shown in Figure 6 (e.g. *s4grid:PowerLimit*). Notice that some meter properties specify a unit of measure (e.g. *om:watt*).

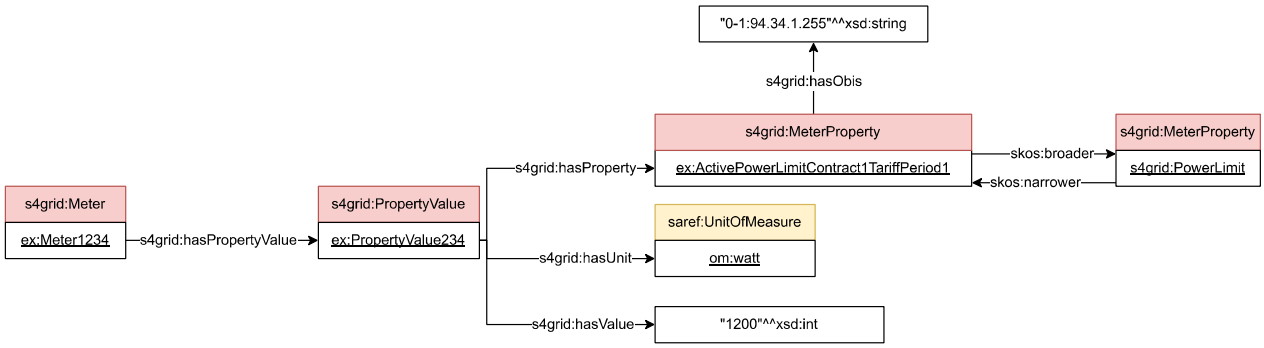

Figure 21: Example of electric grid meter information II

Unlike other SAREF extensions, a meter firmware is not defined by a datatype property. The example presented in Figure 22 depicts a meter firmware (e.g. *ex:ActivePLCFirmware*) which is represented by an OBIS code (*s4grid:hasObis*).

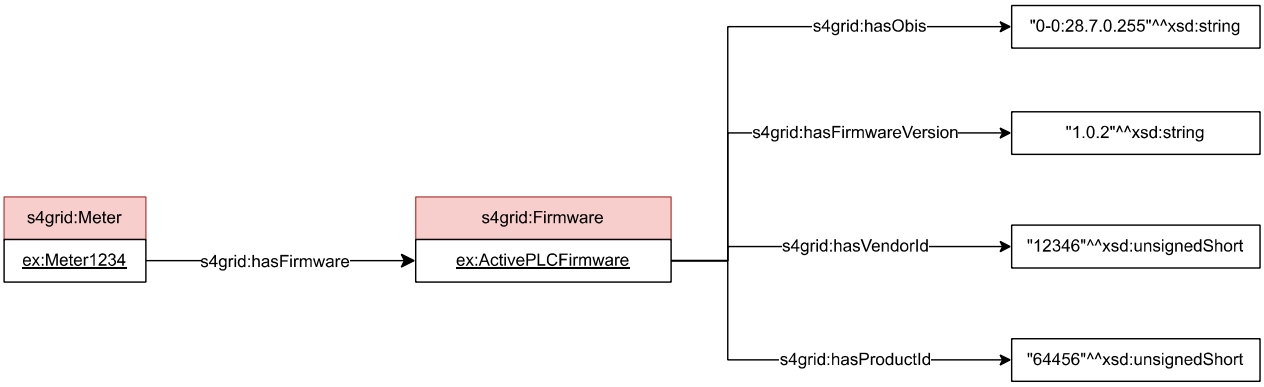

Figure 22: Example of electric grid meter firmware

Figure 23 contains an example of a network interface (*ex:MacAddress1234*) defined for a meter. Moreover, the network interface is represented by an OBIS code (*s4grid:hasObis*).

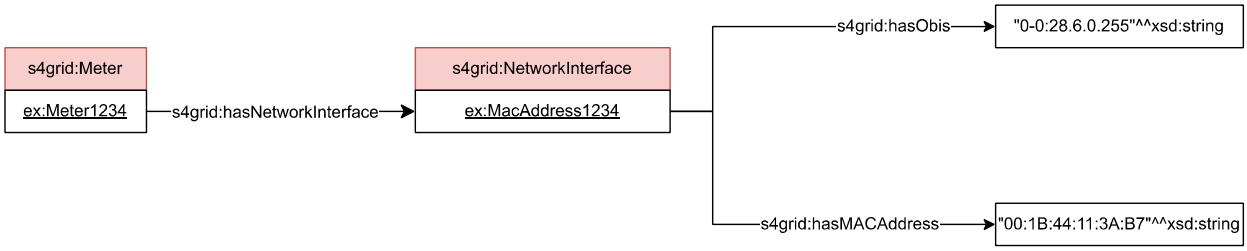

Figure 23: Example of electric grid meter network interface

Figure 24 contains an example of a meter clock (*ex:Clock1234*). Moreover, the clock is represented by an OBIS code (*s4grid:hasObis*). Notice that the clock is not only represented by a time (*s4grid:hasTime*) but is also represented by the time zone in which it is located and how the time is changed.

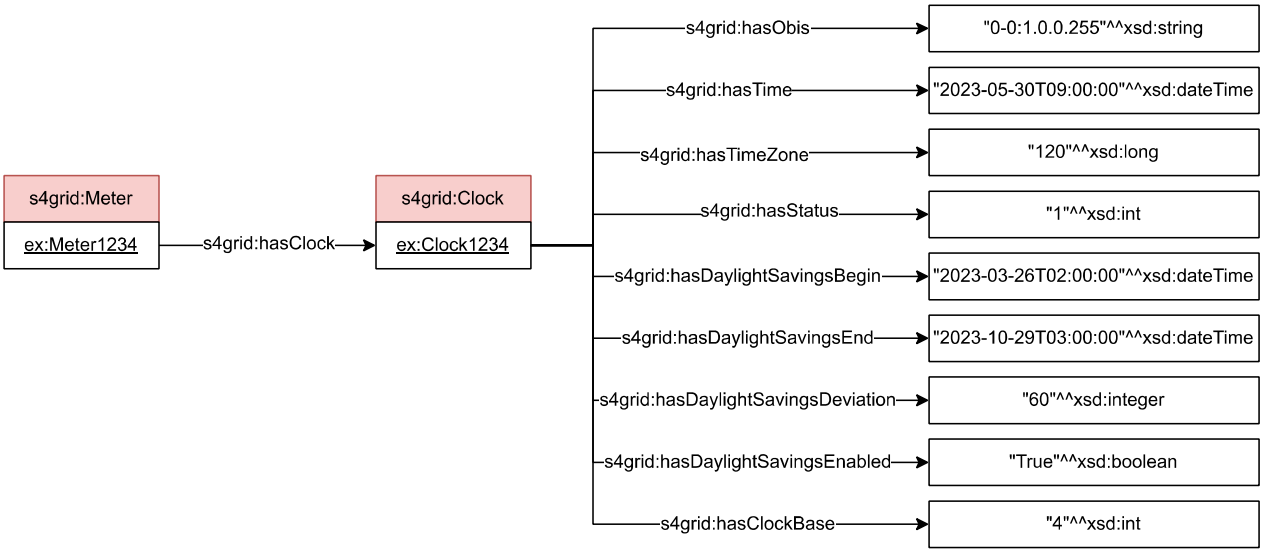

Figure 24: Example of electric grid meter clock

Figure 25 contains an example of a meter breaker state (*ex:CurrentBreakerState*). In this example, it is represented that the meter is physically connected (*s4grid:hasOutputState*), internally connected (*s4grid:hasControlState*), and it can be remotely, manually and locally disconnected (*s4grid:hasControlMode*). Moreover, the breaker state is represented by an OBIS code (*s4grid:hasObis*).

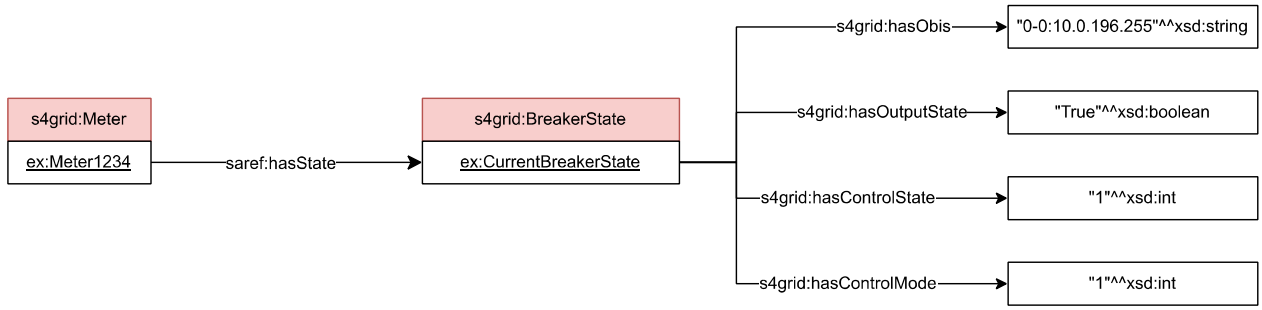

Figure 25: Example of electric grid meter breaker state

Figure 26 contains an example of how scripts (*ex:ConnectionScript* and *ex:DisconnectionScript*) are stored in the meter. A script table (*ex:DisconnectScriptTable*) is needed in order to represent where the scripts are located. Moreover, a single scheduled action (*ex:DisconnectControlScheduler*) is used to represent that a script is going to be executed in a determined date. Moreover, the script table and single scheduled action are represented by an OBIS code (*s4grid:hasObis*).

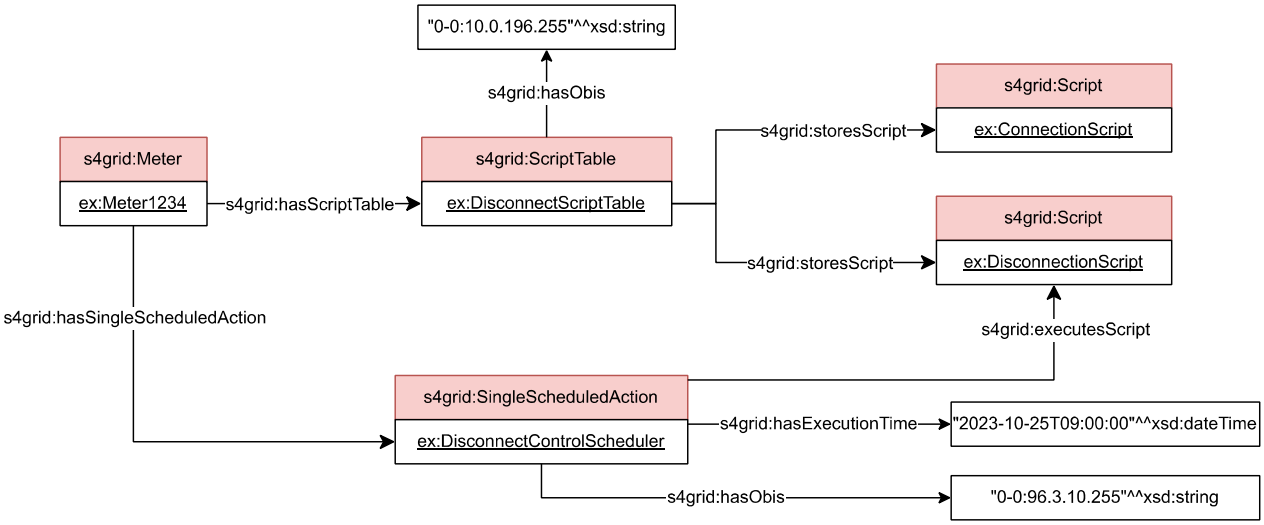

Figure 26: Example of electric grid meter script table and scheduled action

Figure 27 contains an example of how an activity calendar (*ex:ActivityCalendarContract1-1234*) is represented in a meter. This activity calendar is represented by an active season profile (*ex:SeasonActive1*) that is described by the date at which it starts and two regular day profiles: one that describes working days (*ex:WeekDay*) and other that describes weekend days (*ex:WeekendDay*). Each day profile is represented by when a billing period starts each day (*ex:WeekDayTariffPeriod1* and *ex:WeekendDayTariffPeriod1*) and what scripts need to be executed to make the billing (*ex:ResetBillingPeriod1* and *ex:ResetBillingPeriod2*). Additionally, a special day profile (*ex:SpecialFriday1*) defines which days are special (e.g. festive) and, therefore, another tariffication is going to be applied. Moreover, the activity calendar is represented by an OBIS code (*s4grid:hasObis*). Notice that just the active calendar is represented (e.g. the passive calendar associated to the active calendar is not represented) in order to simplify the example.

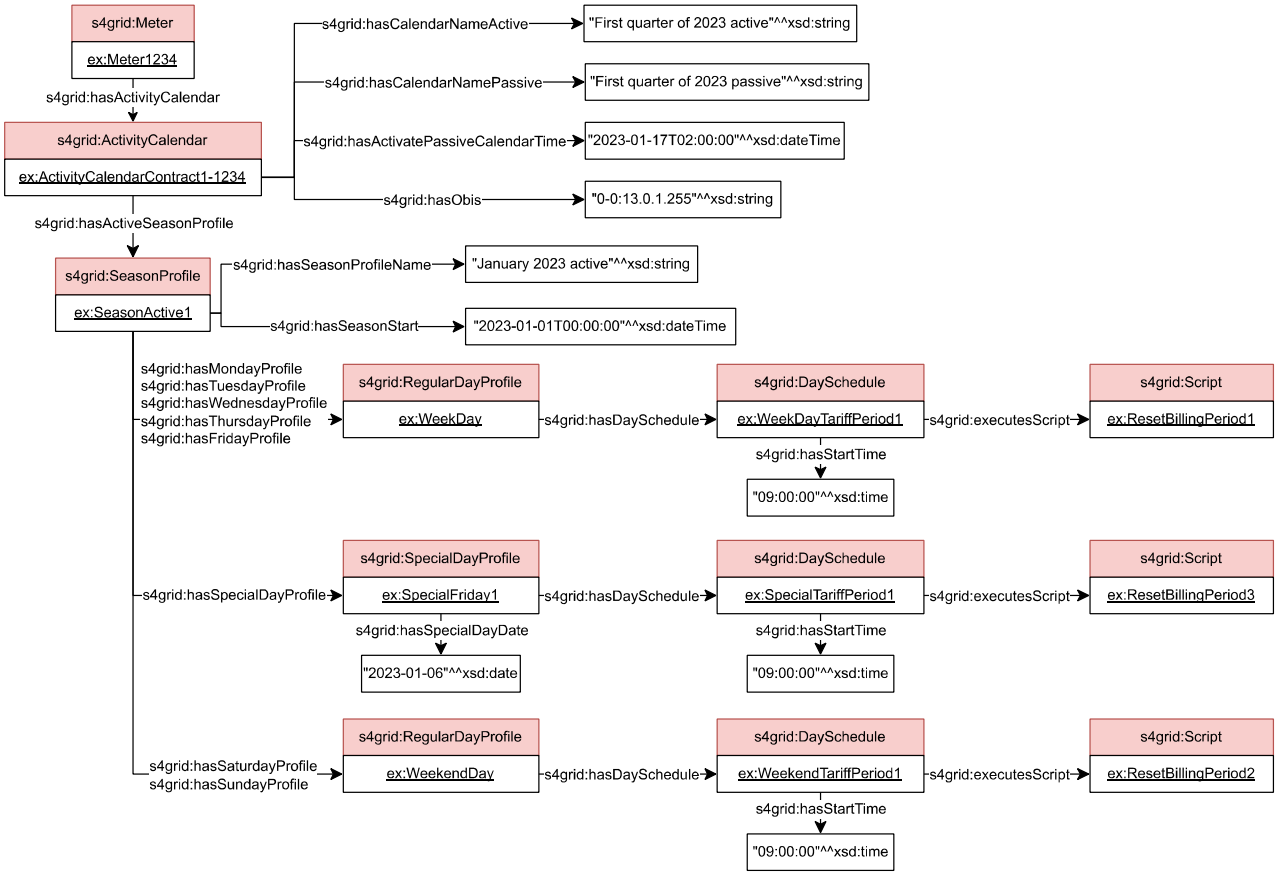

Figure 27: Example of electric grid meter activity calendar

One of the main functions of electric grid meters is to take measures from a power line in order to control what is happening in the electric grid. Figure 28 presents an example of a power line measurement (*ex:TotalIncrementalActiveEnergyImportMeasurement123*) for a power line property (*ex:TotalIncrementalActiveEnergyImport*) that is identified by an OBIS code (*s4grid:hasObis*).

As with meter properties, SAREF4GRID does not aim to provide an exhaustive definition of all the properties defined in the IEC 62056-6-2 [i.4]. Instead, it defines a set of general properties (those shown in Figure 14 and Figure 15, e.g. *s4grid:ActiveEnergy* in the figure) and specific properties can be related to these general properties using the SKOS ontology [i.5]. Using SKOS more specific properties can be defined specifying from which general property they are derived (*skos:narrower*), and which properties belong to a general property (*skos:broader*).

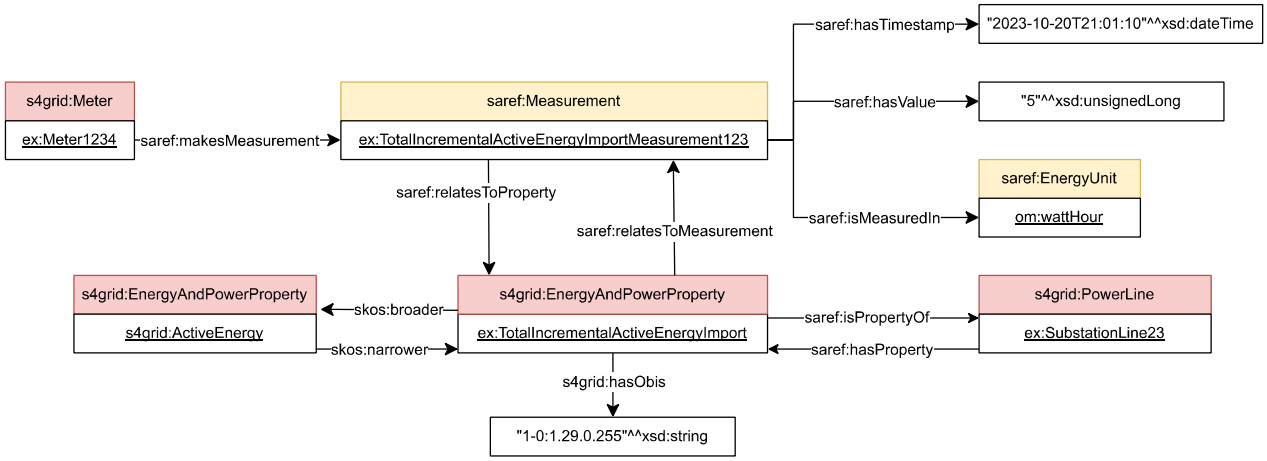

Figure 28: Example of electric grid meter measurements I

Figure 29 presents another example of a power line measurement (*ex:MaximumDemandRegisterImportC1TP1Measurement136*) for a power line property (*ex:MaximumDemandRegisterImportC1TP1*) that is identified by an OBIS code (*s4grid:hasObis*). The power line properties are broader than the properties shown in Figure 14 and Figure 15 (e.g. *s4grid:DemandRegister*).

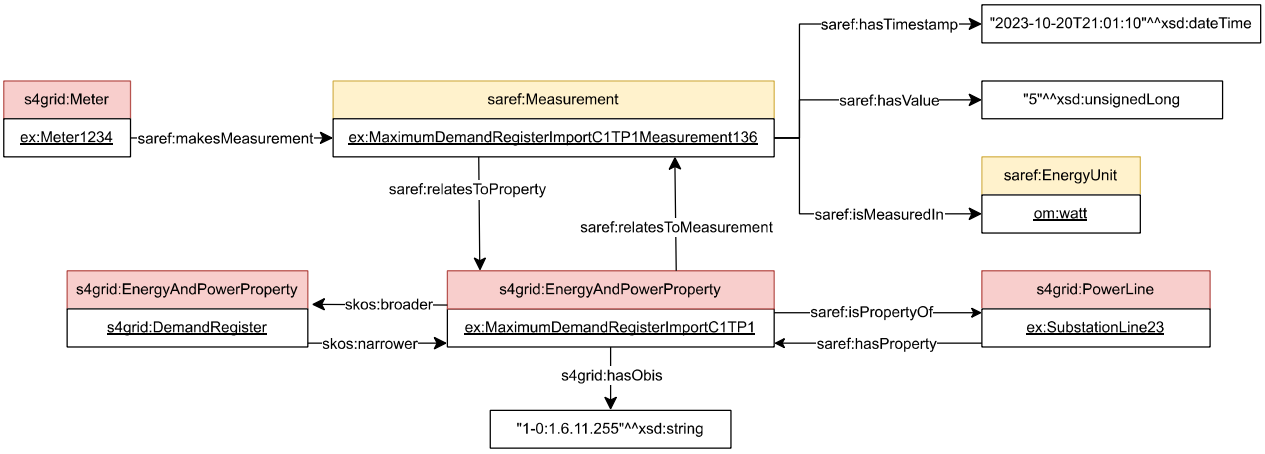

Figure 29: Example of electric grid meter measurements II

Different profile generics can be generated in order to access data groups that are stored in an electric grid meter. Figure 30 presents an example of a profile generic (*ex:IncrementalLoadProfile1234*) whose objective is to store the incremental energy values that a meter measures (*ex:Measurement1* to *ex:Measurement6*) each hour. Additionally, the profile generic stores the clock (*ex:Clock1234*) to indicate the time at which the profile generic captures elements, and the AMR status (*ex:PropertyValue1*) that the meter stores. Moreover, the profile generic is represented by an OBIS code (*s4grid:hasObis*).

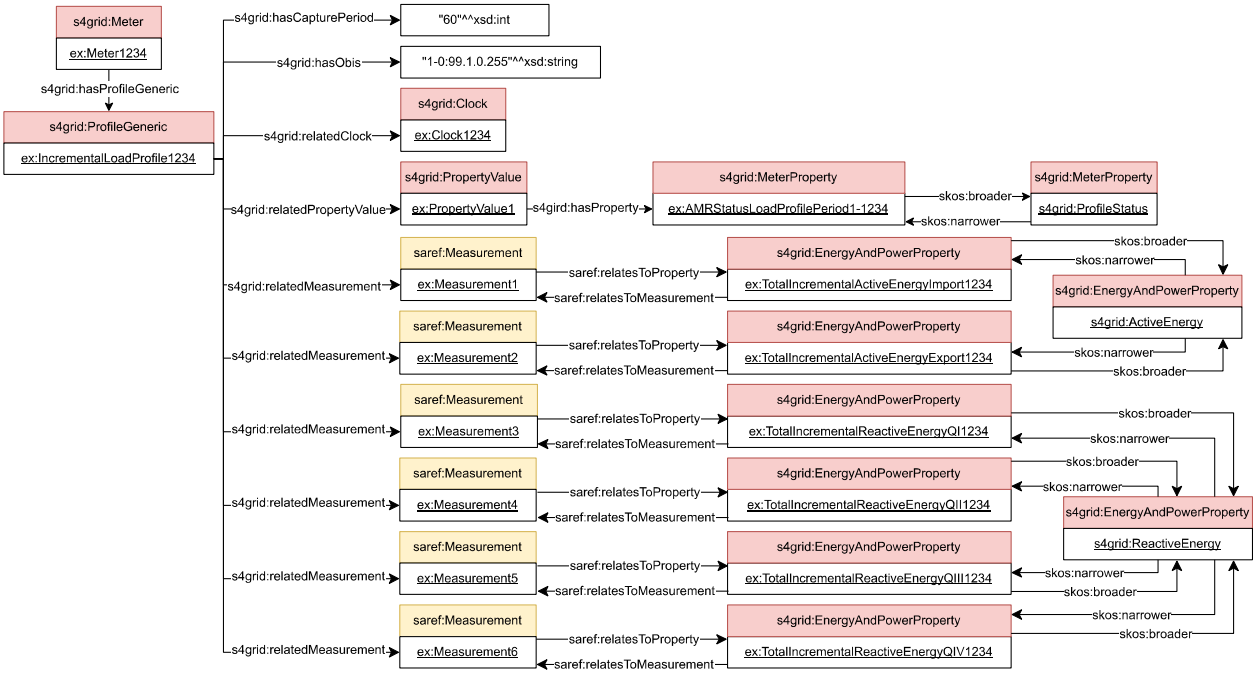

Figure 30: Example of electric grid meter profile generic

Each element that a meter stores can be obtained through a get service. Figure 31 presents an example of how it is specified that a COSEM element is going to be obtained. A get service (*ex:GetServiceIncrementalLoadProfile*) is executed through a get operation (*ex:GetOperationIncrementalLoadProfile*). This operation indicates the desired input (*ex:OperationInputIncrementalLoadProfile*), which in this case specifies the element from which data is going to be retrieved, and the desired output (*ex:OutputIncrementalLoadProfile*), which in this case specifies the data structure that is going to be given.

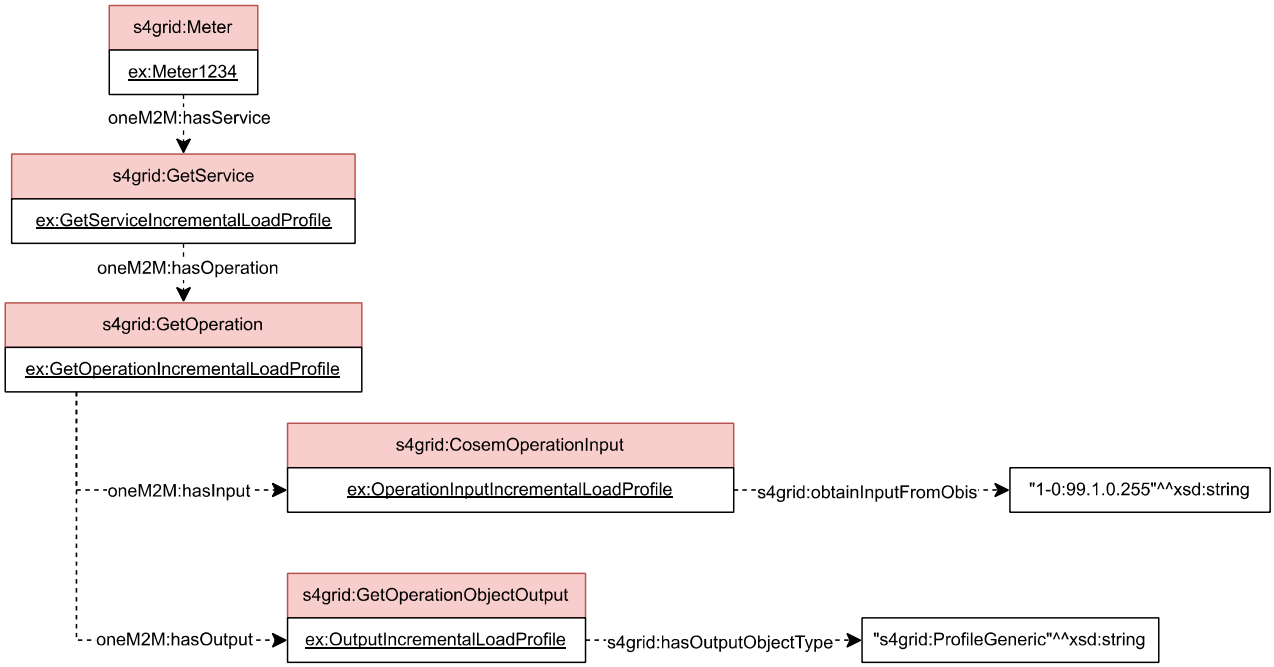

Figure 31: Example of electric grid meter get service I

Figure 32 presents an example of how it is specified that an attribute of a COSEM element is going to be obtained. Notice that in this example, the attribute corresponds to a datatype property of the SAREF4GRID ontology. A get service (*ex:GetServiceCapturePeriodIncrementalLoadProfile*) is executed through a get operation (*ex:GetOperationCapturePeriodIncrementalLoadProfile*). This operation indicates the desired input (*ex:OperationInputCapturePeriodIncrementalLoadProfile*), which in this case specifies the datatype property of the element from which data is going to be retrieved, and the desired output (*ex:OutputCapturePeriodIncrementalLoadProfile*), which in this case specifies the datatype that is going to be given.

Figure 32: Example of electric grid meter get service II

Figure 33 presents an example of how it is specified that an attribute of a COSEM element is going to be obtained. Notice that in this example, the attribute corresponds to an object property of the SAREF4GRID ontology. A get service (*ex:GetServiceDisconnectionScript*) is executed through a get operation (*ex:GetOperationDisconnectionScript*). This operation indicates the desired input (*ex:OperationInputDisconnectionScript*), which in this case specifies the object property of the element from which data is going to be retrieved, and the desired output (*ex:OutputDisconnectionScript*), which in this case specifies the data structure that is going to be given.

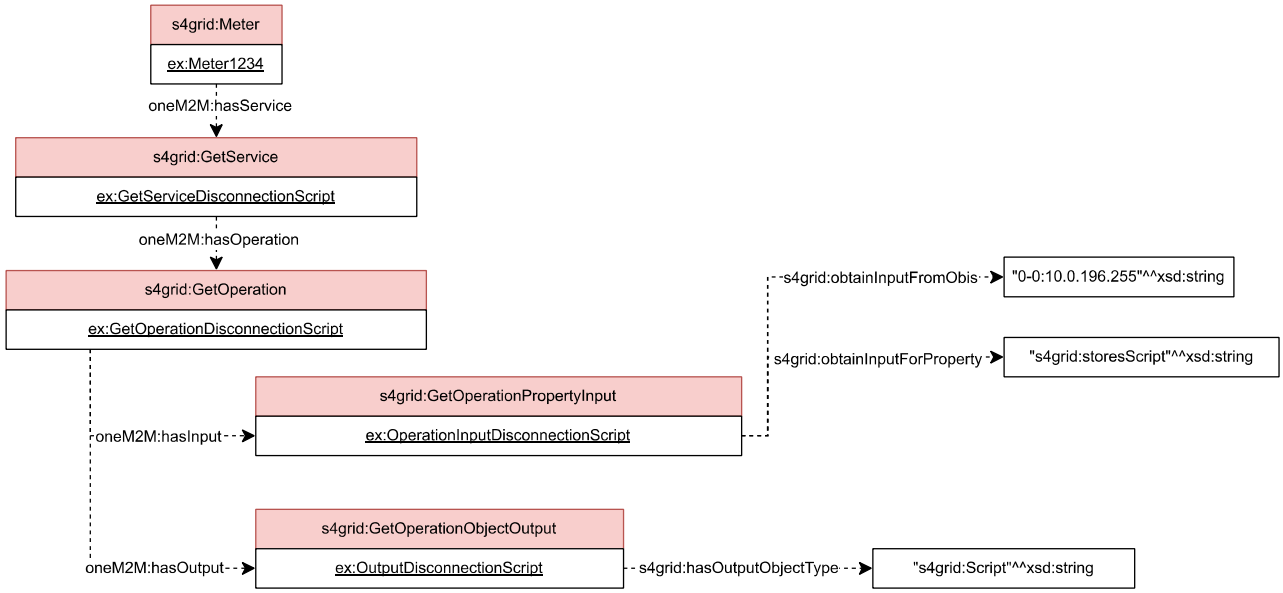

Figure 33: Example of electric grid meter get service III

Property-related services in get operations usually refer to the entire property. However, in the case of certain properties, selective access to only part of the property may be provided. Figure 34 presents an example of how it is specified that a range of values are going to be retrieved from a profile generic. A selective range get service (*ex:GetServiceIncrementalLoadProfileRange*) is executed through a get operation (*ex: GetOperationIncrementalLoadProfileRange*).

This operation indicates the desired input (*ex:OperationInputIncrementalLoadProfileRange*), which in this case specifies the element from which data is going to be retrieved, and the desired output (*ex:OutputIncrementalLoadProfileRange*), which in this case specifies the data structure that is going to be given. Additionally, a selective access with a range descriptor (*ex:RangeDescriptor1*) indicates that just the entries whose value is between 1 and 10 are going to be retrieved.

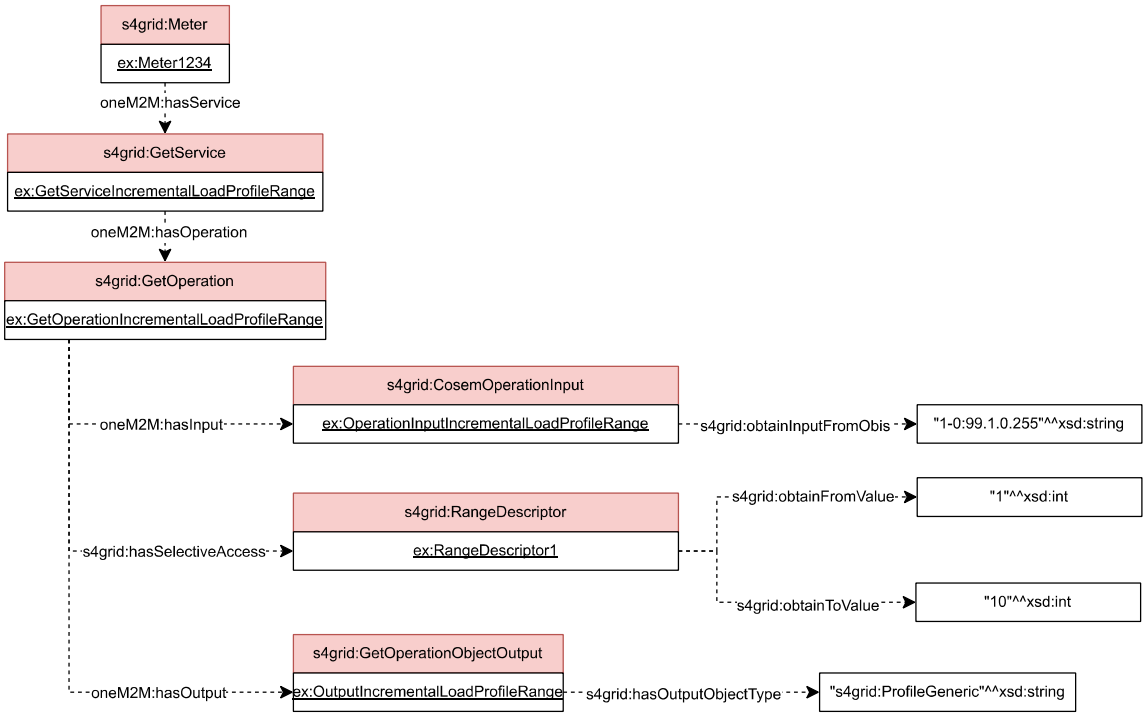

Figure 34: Example of electric grid meter get service IV

Figure 35 presents an example of how it is specified that a range of entries are going to be retrieved from a profile generic. A selective entry get service (*ex:GetServiceIncrementalLoadProfileEntry*) is executed through a get operation (*ex: GetOperationIncrementalLoadProfileEntry*). This operation indicates the desired input (*ex:OperationInputIncrementalLoadProfileEntry*), which in this case specifies the element from which data is going to be retrieved, and the desired output (*ex:OutputIncrementalLoadProfileEntry*), which in this case specifies the data structure that is going to be given. Additionally, a selective access with an entry descriptor (*ex:EntryDescriptor1*) indicates that just the top 10 entries are going to be retrieved.

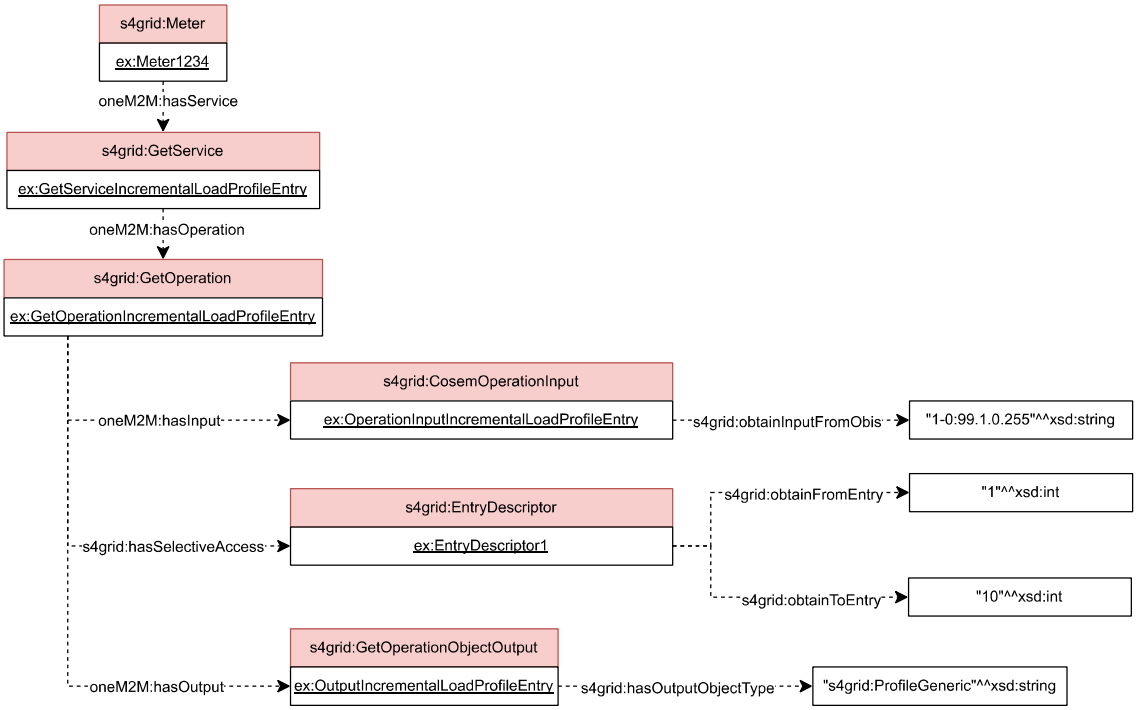

Figure 35: Example of electric grid meter get service V

Each element that a meter stores can be modified through a set service. Figure 36 presents an example of how it is specified that a COSEM element is going to be modified. A set service (*ex:SetMulticastCommunicationIdentifier*) is executed through a set operation (*ex:SetOperationMulticastCommunicationIdentifier*). This operation indicates the desired input (*ex:InputMulticastCommunicationIdentifier*), which in this case specifies the element from which data is going to be modified and the data structure that is going to replace the previous data.

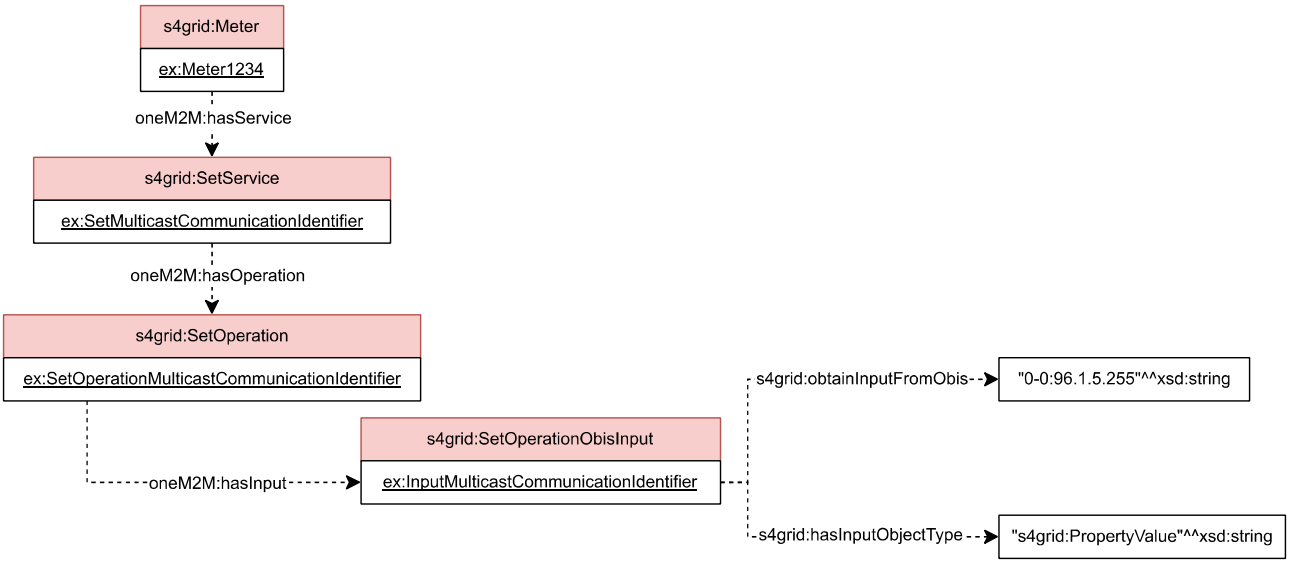

Figure 36: Example of electric grid meter set service I

Figure 37 presents an example of how it is specified that an attribute of a COSEM element is going to be modified. Notice that in this example, the attribute corresponds to a datatype property of the SAREF4GRID ontology. A set service (*ex:SetServiceCapturePeriodIncrementalLoadProfile*) is executed through a set operation (*ex:SetOperationCapturePeriodIncrementalLoadProfile*). This operation indicates the desired input (*ex:InputICapturePeriodIncrementalLoadProfile*), which in this case specifies the datatype property of the element from which data is going to be modified and the data type that is going to replace the previous data.

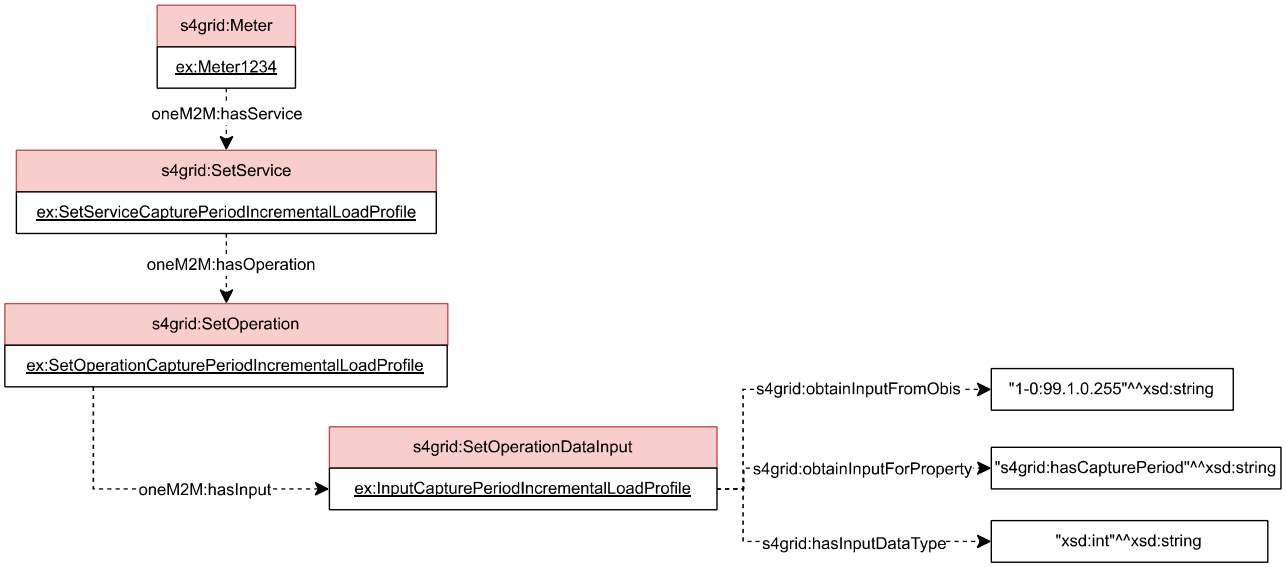

Figure 37: Example of electric grid meter set service II

Figure 38 presents an example of how it is specified that an attribute of a COSEM element is going to be modified. Notice that in this example, the attribute corresponds to an object property of the SAREF4GRID ontology. A set service (*ex:SetServiceDisconnectionScript*) is executed through a set operation (*ex:SetOperationDisconnectionScript*). This operation indicates the desired input (*ex:InputIDisconnectionScript*), which in this case specifies the object property of the element from which data is going to be modified and the data structure that is going to replace the previous data.

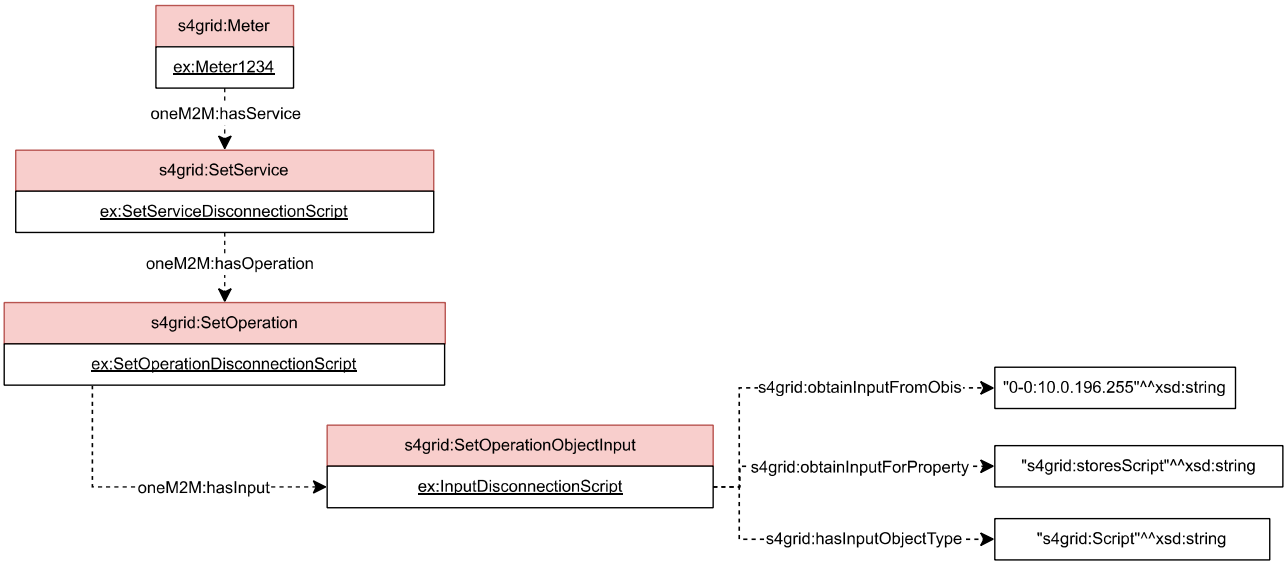

Figure 38: Example of electric grid meter set service III

Each element that a meter stores can be affected through an action service. Figure 39 presents an example of how it is specified that a COSEM element is going to be affected by an action. Notice that in this example the input parameter needed to execute the action is simple (e.g. integer, string, etc.). An action service (*ex:ResetServiceRegister*) is executed through an action operation (*ex:ResetServiceOperation*). This operation indicates the desired input (*ex:ResetServiceOperationInput*), which in this case specifies the element that is going to be affected by the action and the value of the parameter needed by the action.

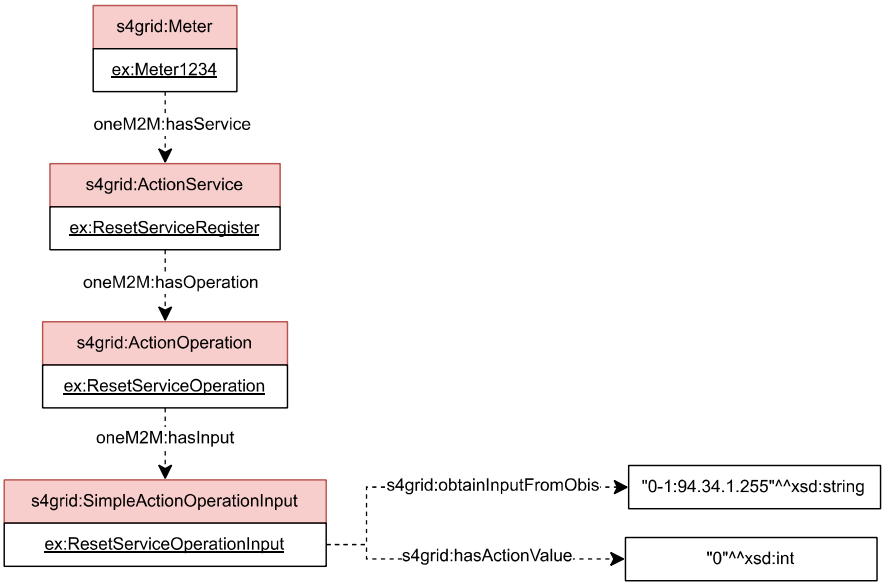

Figure 39: Example of electric grid meter action service I

Figure 40 presents an example of how it is specified that a COSEM element is going to be affected by an action. Notice that in this example the input parameter needed to execute the action is a structure. An action service (*ex:PresetAdjustingTimeServiceClock*) is executed through an action operation (*ex:PresetAdjustingTimeOperation*). This operation indicates the desired input (*ex:PresetAdjustingTime1*), which in this case specifies the element that is going to be affected by the action and the values of the parameter structure needed by the action.

Figure 40: Example of electric grid meter action service II

Figure 41 presents an example of how it is specified that a COSEM element is going to be affected by an action. Notice that in this example the input parameter needed to execute the action is a structure. An action service (*ex:SpecialDayEntryServiceActivityCalendar*) is executed through an action operation (*ex:SpecialDayEntryOperation*). This operation indicates the desired input (*ex:SpecialDayEntry1*), which in this case specifies the element that is going to be affected by the action and the values of the parameter structure needed by the action.

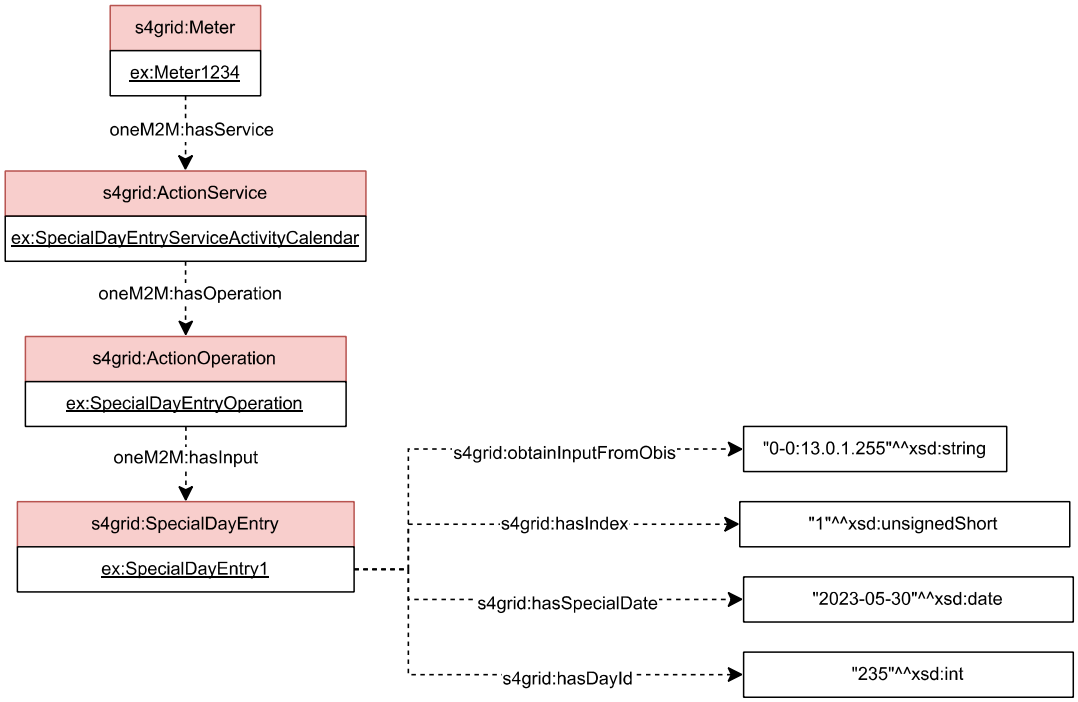

Figure 41: Example of electric grid meter action service III

4.4 Observations

In the following paragraphs, several observations about the SAREF4GRID ontology and its usage are mentioned.

The hierarchies and individuals defined in the extension should not be considered exhaustive. It might be needed to extend the hierarchies and lists of individuals for particular use cases, as well as to specialize some of the defined classes.

The SAREF ontology has been used for the representation of properties. However, the properties model had to be extended to include properties that are not observable by the meter, and therefore, they are not related to a measurement. This has been done by adding the *s4grid:PropertyValue*, which could be moved to SAREF if required by other extensions.

All the structures and individuals have been extracted from the DLMS/COSEM standard. In that standard, all the elements are identified by an OBIS code which is necessary to keep in the ontology.

Annex A (informative): Change history

Date	Version	Information about changes
02-12-2022	0.0.1	Early draft
07-09-2023	0.1.1	Stable draft
16-10-2023	0.2.0	Final draft
November 2023	V1.1.1	First published version

History

Document history		
V1.1.1	November 2023	Publication