ETSI TR 103 783 V1.1.1 (2022-05)

SmartM2M;
SAREF: SDT interoperability
and oneM2M base ontology alignment

Reference
DTR/SmartM2M-103783

Keywords
interoperability, IoT, IoT platforms, oneM2M,
SAREF, semantic, smart lift

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 Report (TR) has been produced by ETSI Technical Committee Smart Machine-to-Machine communications (SmartM2M).

Modal verbs terminology

In the present document "**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 studies the ability to assure full alignment of SAREF and the oneM2M base ontology and to provide guidelines about how devices adopting the oneM2M SDT (Smart Device Template) informational model can interoperate seamlessly with oneM2M devices and systems adopting SAREF and vice versa.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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 TS 103 264 (V3.1.1): "SmartM2M; Smart Applications; Reference Ontology and oneM2M Mapping".

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

[i.3] ETSI TS 118 130: "oneM2M; Ontology based Interworking".

[i.4] oneM2M TS-0023: "SDT based Information Model and Mapping for Vertical Industries".

[i.5] oneM2M TS-0034: "Semantic Support".

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:

CSE	Common Services Entity
IPE	Interworking Proxy application Entity
IRI	Internationalized Resource Identifier

RDF	Resource Description Framework
RDM	Remote Device Management
SAREF	Smart Applications REFerence ontology
SDT	Smart Device Template
SPARQL	SPARQL Protocol and RDF Query Language
XSD	(eXtensible Markup Language) Schema Definition

4 Requirements

4.1 SAREF requirements

The requirements presented in this clause have been derived from the SAREF specification [i.1]. The associated requirements have been grouped in different categories and are presented from Table 1 to Table 6.

Table 1: Requirements for the "Device" category

Id	Competency Question/Statement	Answer
SAREF-1	A device can have a model	
SAREF-2	A device can have a manufacturer	
SAREF-3	A device has at least one function	
SAREF-4	A device can be used for the purpose of offering a commodity	
SAREF-5	A device can measure a property	
SAREF-6	A device may consist of other devices	
SAREF-7	A device can be divided as actuator, sensor, meters and appliance	
SAREF-8	Examples of functions are actuating function, sensing function, metering function and event function	
SAREF-9	Examples of actuating functions are the on off function and the level control function	
SAREF-10	The on off function allows the commands on, off and toggle	
SAREF-11	The level control function allows the commands set level, step up and step down.	
SAREF-12	The sensing function allows the command to get sensed values	
SAREF-13	The metering function allows the commands related to get the current meter value, get the meter data and get the meter history	
SAREF-14	The event function allows the command to notify events	
SAREF-15	A washing machine is a device of type appliance	
SAREF-16	A washing machine accomplishes the task washing	
SAREF-17	A washing machine performs an actuating function of type start-stop	
SAREF-18	A sensor performs a sensing function	
SAREF-19	A temperature sensor consists of a sensor	
SAREF-20	A temperature sensor performs the sensing function	
SAREF-21	A temperature sensor measures temperature	
SAREF-22	A smoke sensor consists of a sensor	
SAREF-23	A smoke sensor performs the sensing and event functions	
SAREF-24	A switch is an actuator	
SAREF-25	A switch performs an actuating function of type on-off and open-close	
SAREF-26	A door switch consists of a switch	
SAREF-27	A door switch is an actuator	
SAREF-28	A door switch performs the open-close function	
SAREF-29	A dimmer lamp is an actuator	
SAREF-30	A dimmer lamp performs an actuating level control function	
SAREF-31	A meter performs a metering function	
SAREF-32	An energy meter consists of a meter	
SAREF-33	An energy meter performs a metering function	
SAREF-34	An energy meter measures energy	

Table 2: Requirements for the "Command" category

Id	Competency Question/Statement	Answer
SAREF-35	A function has at least one command	
SAREF-36	The on, off and toggle commands are associated to the on-off function	
SAREF-37	The set level, step up and step-down commands are associated to the level control function	
SAREF-38	The get command is associated to the sensing function	
SAREF-39	The get current meter value, get meter data and get meter history commands are associated to the metering function	
SAREF-40	The notify command is associated to the event function	
SAREF-41	A command can act upon a state	

Table 3: Requirements for the "State" category

Id	Competency Question/Statement	Answer
SAREF-42	A device can be found in a state	

Table 4: Requirements for the "Service" category

Id	Competency Question/Statement	Answer
SAREF-43	A device offers a service	
SAREF-44	A service represents at least one function	
SAREF-45	A service is offered by at least one device	

Table 5: Requirements for the "Profile" category

Id	Competency Question/Statement	Answer
SAREF-46	A device can have a profile	
SAREF-47	A profile is linked to a certain property or commodity	
SAREF-48	A profile can be calculated over a time span	
SAREF-49	A profile can be associated to some costs	

Table 6: Requirements for the "Measurement" category

Id	Competency Question/Statement	Answer
SAREF-50	A measurement describes a measure of a quantity	
SAREF-51	A measurement is related to a property	
SAREF-52	A measurement is described according to a given unit of measure	
SAREF-53	A measurement can have a timestamp	
SAREF-54	A property is related to a measurement	
SAREF-55	A feature of interest can be associated to a measurement	
SAREF-56	Examples of properties are power and energy	
SAREF-57	The power unit is a unit of measurement	
SAREF-58	The energy unit is a unit of measurement	

4.2 oneM2M requirements

The requirements presented in this clause have been derived from ETSI TS 118 112 [i.2]. The associated requirements have been grouped in different categories and are presented from Table 7 to Table 17.

Table 7: Requirements for the "Thing" category

Id	Competency Question/Statement	Answer
oneM2M-1	A thing is an entity that can be identified in the oneM2M system	
oneM2M-2	A thing may have properties	
oneM2M-3	A thing may have relations to other things	
oneM2M-4	A thing property denotes a property of a thing	
oneM2M-5	A thing can be described with (the values of) ThingProperties	

Table 8: Requirements for the "Variable" category

Id	Competency Question/Statement	Answer
oneM2M-6	A Variable constitutes a super class to the following classes: ThingProperty, OperationInput, OperationOutput, InputDataPoint, OutputDataPoint, SimpleTypeVariable, StructuredTypeVariable	
oneM2M-7	A variable is an entity that has some data (e.g. integers, text, etc. or structured data)	
oneM2M-8	The data of the variable can describe some real-world aspects	
oneM2M-9	The data of the variable can have metadata	
oneM2M-10	A variable can consist of (sub-) variables	
oneM2M-11	A variable conversion represents a conversion rule from the value range of one variable into the value range of another variable	

Table 9: Requirements for the "Metadata" category

Id	Competency Question/Statement	Answer
oneM2M-12	Metadata contain data about a variable	
oneM2M-13	Metadata contain data about an aspect	

Table 10: Requirements for the "Device" category

Id	Competency Question/Statement	Answer
oneM2M-14	A device can accomplish a particular task via the functions	
oneM2M-15	A device can interact electronically with its environment via a network	
oneM2M-16	A device can have services	
oneM2M-17	A device interacts through the data points and/or operations of its services	
oneM2M-18	A device performs one or more functions	
oneM2M-19	Functions are exposed in the network as services of the device	
oneM2M-20	A device can be composed of several (sub-)devices	
oneM2M-21	Each device needs to be individually addressable in the network	

Table 11: Requirements for the "Function" category

Id	Competency Question/Statement	Answer
oneM2M-22	A controlling functionality is a type of function	
oneM2M-23	A measuring functionality is a type of function	
oneM2M-24	A device can be designed to perform more than one function	
oneM2M-25	A function refers to a certain aspect	
oneM2M-26	ControllingFunction is a sub-class of Function that only controls/influences real world Aspects that the Function relates to	
oneM2M-27	MeasuringFunction (Class: MeasuringFunction) is a sub-class of Function that only measures/senses real world aspects that the function relates to	

Table 12: Requirements for the "Aspect" category

Id	Competency Question/Statement	Answer
oneM2M-28	An aspect can be a physical or non-physical entity	
oneM2M-29	An aspect describes the real-world aspect that a function relates to	
oneM2M-30	Aspect can also be used to describe a quality or kind of OperationInput or OperationOutput variables	

Table 13: Requirements for the "Command" category

Id	Competency Question/Statement	Answer
oneM2M-31	A Command (Class: Command) represents an action that can be performed to support the Function	
oneM2M-32	An Operation exposes a Command to the network	
oneM2M-33	OperationInput and OperationOutput of the related Operation can parameterize the command	

Table 14: Requirements for the "Service" category

Id	Competency Question/Statement	Answer
oneM2M-34	A service exposes a function	
oneM2M-35	A service can represent one or more functions	
oneM2M-36	A service can have subservices	
oneM2M-37	A service may have operations	

Table 15: Requirements for the "Operation" category

Id	Competency Question/Statement	Answer
oneM2M-38	A service uses the operations	
oneM2M-39	An operation produces outputs	
oneM2M-40	An operation can have inputs	
oneM2M-41	GET_InputDataPoint is a sub-class of operation	
oneM2M-42	SET_OutputDataPoint is a sub-class of operation	
oneM2M-43	OperationInput describes the type of input of an operation to a service of the device	
oneM2M-44	An operation can have multiple operationOutputs	
oneM2M-45	An operation can have multiple operationInputs	
oneM2M-46	OperationOutput describes the type of output of an operation from a service of the device	

Table 16: Requirements for the "Area Network" category

Id	Competency Question/Statement	Answer
oneM2M-47	An Area Network is characterized by physical properties, its communication protocol and a profile	

Table 17: Requirements for the "Interworked Device" category

Id	Competency Question/Statement	Answer
oneM2M-48	An interworked device is part of an area network	

5 Conformance tests

5.1 SAREF tests

This clause presents the set of tests obtained from the SAREF requirements described in clause 4.1. The associated tests have been grouped in different categories and are presented from Table 18 to Table 23.

Table 18: Requirements for the "Device" category

Requirement Id	Test
SAREF-1	Device subClassOf hasModel only string
SAREF-2	Device subClassOf hasManufacturer only string
SAREF-3	Device subclassOf hasFunction minimum 1 Function
SAREF-4	Device isUsedFor Commodity
SAREF-5	Device measuresProperty Property
SAREF-6	Device consistsOf Device
SAREF-7	Actuator subclassOf Device, Sensor subclassOf Device, Meter subclassOf Device, Appliance subclassOf Device
SAREF-8	ActuatingFunction subclassOf Function, SensingFunction subclassOf Function, MeteringFunction subclassOf Function, EventFunction subclassOf Function
SAREF-9	OnOffFunction subclassOf ActuatingFunction, setLevelControlFunction subclassOf ActuatingFunction
SAREF-10	OnOffFunction hasCommand OnCommand, OnOffFunction hasCommand OffCommand, OnOffFunction hasCommand ToggleCommand
SAREF-11	LevelControlFunction hasCommand setLevelDownCommand, OnOffFunction hasCommand setUpCommand, OnOffFunction hasCommand setDownCommand
SAREF-12	SensingFunction subclassOf hasCommand getCommand
SAREF-13	MeteringFunction hasCommand getCurrentMeterValueCommand, MeteringFunction hasCommand getMeterDataCommand, MeteringFunction hasCommand getMeterHistoryCommand
SAREF-14	EventFunction hasCommand NotifyCommand
SAREF-15	WashingMachine subclassOf Appliance
SAREF-16	WashingMachine accomplishes Washing
SAREF-17	WashingMachine hasFunction StartStopFunction
SAREF-18	Sensor hasFunction SensingFunction
SAREF-19	TemperatureSensor consistsOf Sensor
SAREF-20	TemperatureSensor hasFunction SensingFunction
SAREF-21	TemperatureSensor measuresProperty Temperature
SAREF-22	SmokeSensor consistsOf Sensor
SAREF-23	SmokeSensor hasFunction SensingFunction, SmokeSensor hasFunction EventFunction
SAREF-24	Switch subclassOf Actuator
SAREF-25	Switch hasFunction OnOffFunction, Switch hasFunction OpenCloseFunction
SAREF-26	DoorSwitch consistsOf Switch
SAREF-27	DoorSwitch subclassOf Actuator
SAREF-28	DoorSwitch hasFunction OpenCloseFunction
SAREF-29	DimmerLamp subclassOf Actuator
SAREF-30	DimmerLamp hasFunction LevelControlFunction
SAREF-31	Meter hasFunction MeteringFunction
SAREF-32	EnergyMeter consistsOf Meter
SAREF-33	EnergyMeter hasFunction MeteringFunction
SAREF-34	EnergyMeter measuresProperty Energy

Table 19: Requirements for the "Command" category

Requirement Id	Test
SAREF-35	Function subclassOf hasCommand minimum 1 Command
SAREF-36	OffCommand isCommandOf OnOffFunction, OnCommand isCommandOf OnOffFunction, ToggleCommand isCommandOf OnOffFunction
SAREF-37	SetLevelCommand isCommandOf LevelControlFunction, SetUpCommand isCommandOf LevelControlFunction, SetDownCommand isCommandOf LevelControlFunction
SAREF-38	GetCommand isCommandOf SensingFunction
SAREF-39	GetCurrentMeterValueCommand isCommandOf MeteringFunction, GetMeterDataCommand isCommandOf MeteringFunction, GetMeterHistoryCommand isCommandOf MeteringFunction
SAREF-40	NotifyCommand isCommandOf EventFunction
SAREF-41	Command actsUpon State

Table 20: Requirements for the "State" category

Requirement Id	Test
SAREF-42	Device hasState State

Table 21: Requirements for the "Service" category

Requirement Id	Test
SAREF-43	Device offers Service
SAREF-44	Service subclassOf represents minimum 1 Function
SAREF-45	Service subclassOf isOfferedBy minimum 1 Device

Table 22: Requirements for the "Profile" category

Requirement Id	Test
SAREF-46	Device hasProfile Profile
SAREF-47	Profile isAbout Property, Profile isAbout Commodity
SAREF-48	Profile subclassOf hasTime only Time
SAREF-49	Profile subclassOf hasPrice only Price

Table 23: Requirements for the "Measurement" category

Requirement Id	Test
SAREF-50	Measurement subclassOf hasValue some float
SAREF-51	Measurement relatesToProperty Property
SAREF-52	Measurement isMeasuredIn UnitOfMeasure
SAREF-53	Measurement subclassOf hasTimestamp only datetime
SAREF-54	Property relatesToMeasurement
SAREF-55	FeatureOfInterest hasMeasurement Measurement
SAREF-56	Power subclassOf Property, Energy subclassOf Property
SAREF-57	PowerUnit subclassOf UnitOfMeasurement
SAREF-58	EnergyUnit subclassOf UnitOf Measurement

5.2 oneM2M tests

This clause presents the set of tests obtained from the oneM2M requirements described in clause 4.2. The associated tests have been grouped in different categories and are presented from Table 24 to Table 34.

Table 24: Requirements for the "Thing" category

Requirement Id	Test
oneM2M-1	Thing type Class
oneM2M-2	Thing hasThingProperty ThingProperty
oneM2M-3	Thing hasThingRelation Thing
oneM2M-4	Thing hasThingProperty ThingProperty
oneM2M-5	Thing hasThingProperty ThingProperty

Table 25: Requirements for the "Variable" category

Requirement Id	Test
oneM2M-6	ThingProperty subclassOf Variable, OperationInput subclassOf Variable, OperationOutput subclassOf Variable, InputDataPoint subclassOf Variable, OutputDataPoint subclassOf Variable, SimpleTypeVariable subclassOf Variable, StructuredTypeVariable subclassOf Variable
oneM2M-7	Variable subclassOf hasValue some string
oneM2M-8	Variable describes Aspect
oneM2M-9	Variable hasMetadata Metadata
oneM2M-10	StructuredTypeVariable hasSubstructure Variable
oneM2M-11	Variable hasConversion VariableConversion

Table 26: Requirements for the "Metadata" category

Requirement Id	Test
oneM2M-12	Variable hasMetadata Metadata
oneM2M-13	Aspect hasMetadata Metadata

Table 27: Requirements for the "Device" category

Requirement Id	Test
oneM2M-14	Device hasFunction Function
oneM2M-15	InterworkedDevice isPartOf AreaNetwork, InterworkedDevice subclassOf Device
oneM2M-16	Device hasService Service, Device subclassOf Thing
oneM2M-17	Device hasService Service
oneM2M-18	Device hasFunction Function
oneM2M-19	Device offers Service, Service represents Function
oneM2M-20	Device consistsOf Device
oneM2M-21	Device type Class

Table 28: Requirements for the "Function" category

Requirement Id	Test
oneM2M-22	ActuatingFunction subClassOf Function
oneM2M-23	SensingFunction subClassOf Function
oneM2M-24	Device subclassOf hasFunction minimum 1 Function
oneM2M-25	Function refersTo Aspect
oneM2M-26	ActuatingFunction subClassOf Function
oneM2M-27	SensingFunction subClassOf Function

Table 29: Requirements for the "Aspect" category

Requirement Id	Test
oneM2M-28	Aspect type Class
oneM2M-29	Function refersTo Aspect
oneM2M-30	Variable describes Aspect

Table 30: Requirements for the "Command" category

Requirement Id	Test
oneM2M-31	Function hasCommand Command
oneM2M-32	Operation exposesCommand Command
oneM2M-33	Command hasInput OperationInput, Command hasOutput OperationOutput

Table 31: Requirements for the "Service" category

Requirement Id	Test
oneM2M-34	Service exposesFunction Function
oneM2M-35	Service subClassOf exposesFunction minimum 1 Function
oneM2M-36	Service hasSubService Service
oneM2M-37	Service hasOperation Operation

Table 32: Requirements for the "Operation" category

Requirement Id	Test
oneM2M-38	Service hasOperation Operation
oneM2M-39	Operation hasOutput OperationOutput
oneM2M-40	Operation hasInput OperationInput
oneM2M-41	GET_InputDataPoint subclassOf Operator
oneM2M-42	SET_InputDataPoint subclassOf Operator
oneM2M-43	Operation hasInput OperationInput
oneM2M-44	Operation subClassOf hasOutput only OperationOutput
oneM2M-45	Operation subClassOf hasInput only OperationInput
oneM2M-46	Operation hasOutput OperationOutput

Table 33: Requirements for the "Area Network" category

Requirement Id	Test
oneM2M-47	AreaNetwork subclassOf netTechnologyPhysicalStandard only string, AreaNetwork subclassOf netTechnologyProfile only string, AreaNetwork subclassOf netTechnologyCommunicationProtocol only string

Table 34: Requirements for the "Interworked Device" category

Requirement Id	Test
oneM2M-48	InterworkedDevice isPartOf AreaNetwork

6 Conformance testing results

Figure 1 and Figure 2 present the results of the conformance testing between SAREF [i.1] and the oneM2M Base Ontology [i.2]. The first figure presents an overview of SAREF highlighting in green those terms that are also shared by the oneM2M Base Ontology and, similarly, the second figure presents an overview of the oneM2M Base Ontology highlighting in green those terms that are also shared by the SAREF.

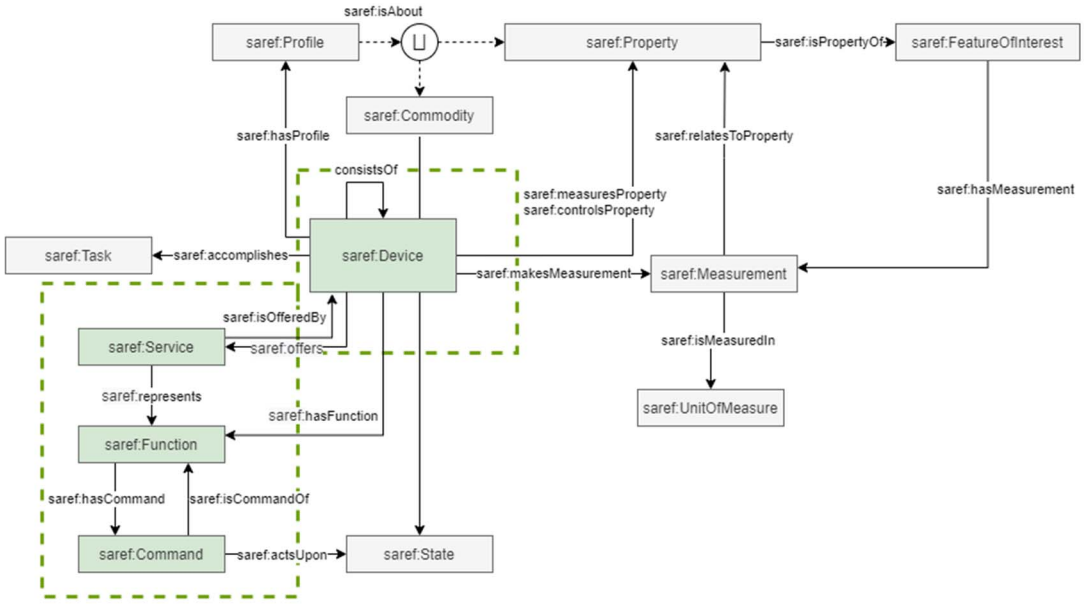

Figure 1: SAREF terms shared by the oneM2M Base Ontology

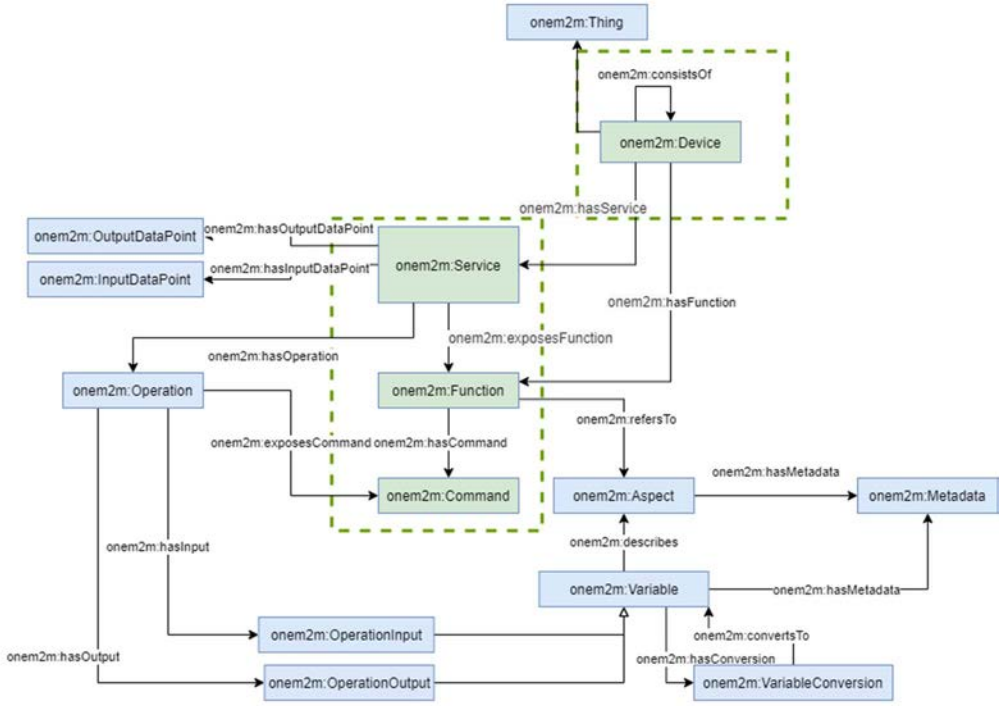

Figure 2: oneM2M Base Ontology terms shared by SAREF

Figure 3 shows a summary of the terms shared by both ontologies. It can be seen that the concepts of Device and Service play a major role in both ontologies. Furthermore, the related concepts of Function and Command also appear in both ontologies.

As relevant as the shared terms are those terms that are not shared by the ontologies, because they indicate potential combined uses of the ontologies or extensions to them. On the one hand, SAREF enables a more detailed description of devices and also allows representing device measurements, which are not covered by the oneM2M Base Ontology. On the other hand, the oneM2M Base Ontology allows describing services in detail, an aspect that is not covered in SAREF.

Furthermore, the conformance testing indicated that there were no conflicts between both ontologies, so they can complement each other using as common grounds the shared terms. These shared terms represent the common knowledge between both ontologies that is the basis for their alignment.

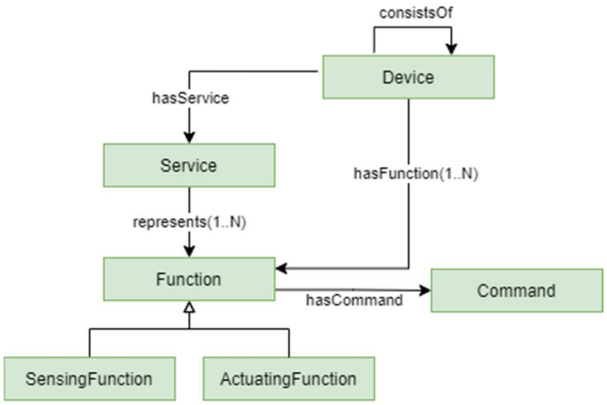

Figure 3: Overview of the shared terms between SAREF and the oneM2M Base Ontology

7 Interworking SAREF devices and SDT devices

7.0 Introduction

As described in ETSI TS 118 112 [i.2], oneM2M Base Ontology, ontologies are used in oneM2M to provide syntactic and semantic interoperability with external systems. Syntactic interoperability, as defined in ETSI TS 118 112 [i.2], is intended to support the allocation of resources to model an external device by an IPE (see clause 7.2 for the mapping of a washing machine). This enables oneM2M entities to access (read, write, and discover) the functionality and services of the device modelled. Semantic interoperability, as defined in ETSI TS 118 112 [i.2], describes functions provided by oneM2M compliant devices and how to access those functions.

The oneM2M ontology supports being able to issue **generic queries to discover devices** and **specific queries to determine what oneM2M resources need to be accessed to use those devices**.

Devices can be modelled using methods defined by oneM2M or custom methods. In ETSI TS 118 130 [i.3] modelling a device in oneM2M is defined based on the mapping of an ontology to define a resource tree structure for the device. In oneM2M TS-0023 [i.4] predefined resource tree structures are defined for a variety of devices across multiple domains. And finally, using oneM2M a device can be modelled using a custom resource tree structure. In clause 7.1, a washing machine is modelled using each of these approaches. Then, the information needed to make these devices interoperable is described in clause 7.2.

The example is a (simplified) washing machine:

- The washing machine has been manufactured by manufacturer **XYZ**.
- XYZ describes this type of washing machine as "`Very cool Washing Machine`".
- The model of the type of washing machine is **XYZ_Cool.**
- The state of the washing machine can take the values "WASHING" or "STOPPED" or "ERROR".
- The washing machine supports three commands: **ON, OFF, Toggle.**
- The washing machine is located in **My_Bathroom.**

7.1 Washing machine use case

7.1.1 Device models using the Smart Device Template

oneM2M TS-0023 [i.4] has multiple device specific domains. The Home Domain contains a deviceClothesWasher model that aligns with the device that it has to be modelled. The resource tree structure of the deviceClothesWasher is shown below (in Figure 4 the elements in bold are required for a compliant SDT model). There are many more potential services exposed in this model than our example simplified washing machine provides.

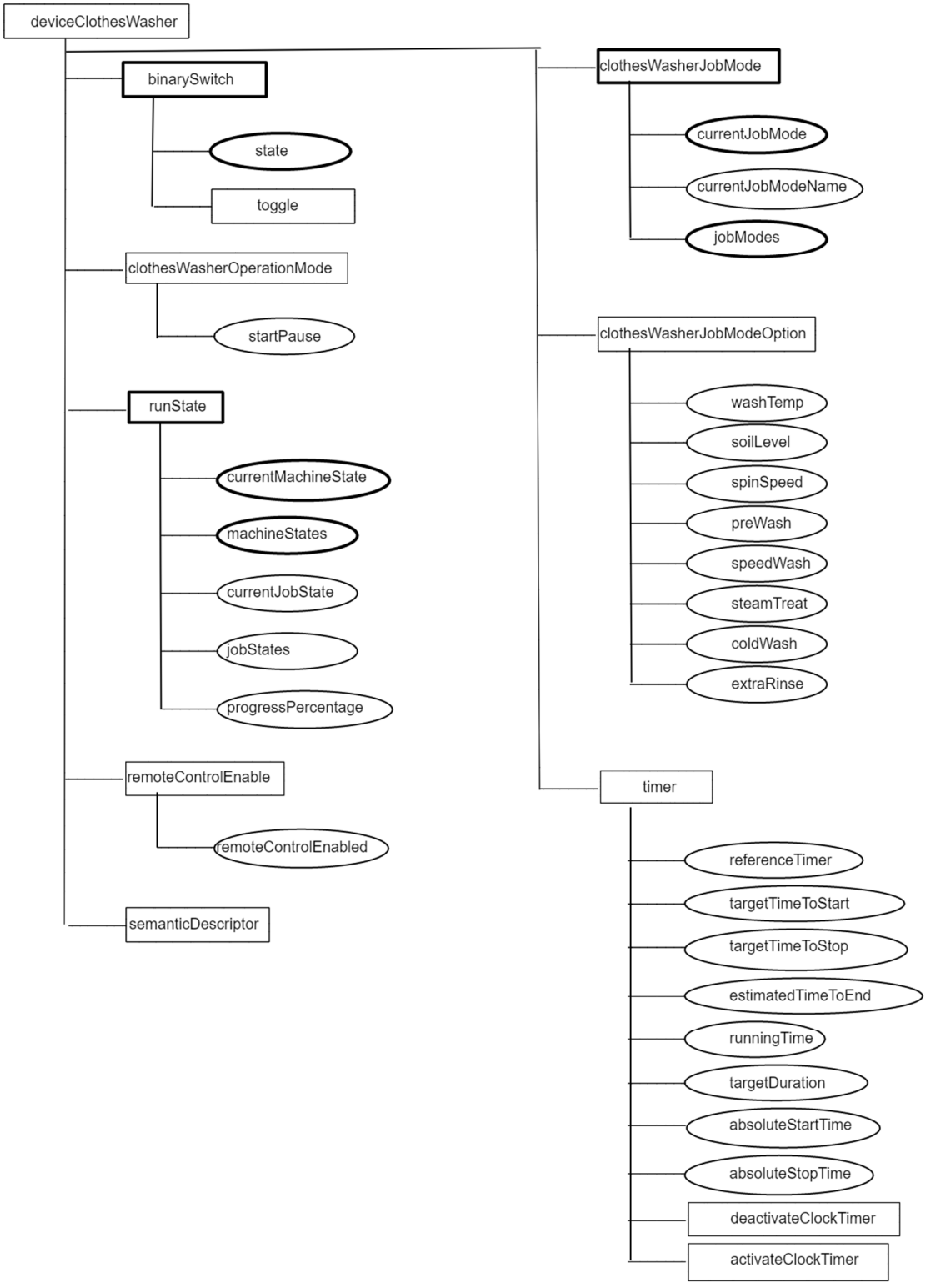

Figure 4: SDT washing machine model

When using a SDT model from oneM2M TS-0023 [i.4] to represent a physical device it is necessary to map the functionality of the device to be modelled with the existing modules defined for the SDT device.

Table 35: Mapping between modelled device functionalities and existing SDT ones

Meta-Data	Device Value	SDT modelling
Manufacturer	XYZ	The SDT model captures this information in a dmDeviceInfo ModuleClass.
Manufacturer description	"Very cool Washing Machine"	The SDT model captures this information in dmDeviceInfo ModuleClass.
Model Type	XYZ_Cool	The SDT model captures this information in a dmDeviceInfo ModuleClass.
Supported Commands	ON_Command OFF_Command Toggle_Command	The SDT model enables the ON and OFF commands using the *state* attribute of the *binarySwitch* ModuleClass. The Toggle command is supported by the *toggle* ActionModule.
State	"WASHING" "STOPPED" "ERROR"	The SDT model offers runState ModuleClass which supports more enumerations that indicated by our product.
Location	My_Bathroom	The SDT model does not have an attribute specifically for Location.

7.1.2 Device models using the SAREF ontology

A SAREF description of the washing machine is mapped to the resource structure shown in Figure 5 using the rules described in ETSI TS 118 130 [i.3]. A complete derivation of this example is shown in clause B.1.3.3 of ETSI TS 118 112 [i.2].

The description of the (simplified) washing machine using SAREF ontology is expanded upon here:

- The state of the washing machine is given by the state: **WashingMachineStatus** that can take the values "WASHING" or "STOPPED" or "ERROR".
- The washing machine has an actuating function: **StartStopFunction** which has three commands:
  - **ON_Command**
  - **OFF_Command**
  - **Toggle_Command**
- The related service of the washing machine that represents that actuating function is of class: **saref:Service**. It has:
  - an InputDataPoint: **BinaryInput** (to expose commands ON_Command and OFF_Command); and
  - an Operation: **ToggleBinary** (to expose command Toggle_Command).
- The washing machine has also a metering function: **MonitoringFunction** that sets the WashingMachineStatus.
- This WashingMachineStatus is provided as the OutputDataPoint of a service **MonitorService** which exposes the MonitoringFunction to the network.
- The washing machine is located at **My_Bathroom.**

It is worthwhile to point out that "InputDataPoint", "OutputDataPoint" and "Operation" are not specified in SAREF, they are classes of the oneM2M Base Ontology. They are included here because, according to ETSI TS 118 112 [i.2], they are needed to properly generate the resource tree structure, shown in Figure 5.

The set of RDF triples is not complete according to the procedures defined in ETSI TS 118 130 [i.3]. Only the triples needed to describe the resource tree for the washing machine use case are included. Additional triples are needed to define the XSD schema for the <flexContainer> resources that will model this device; those are excluded here for the sake of clarity.

```
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#> .
@prefix oneM2M: <http://www.onem2m.org/ontology/Base_Ontology/> .
@prefix saref: <https://saref.etsi.org/core/> .
@prefix s4bldg: <https://saref.etsi.org/saref4bldg/> .
@prefix sn:    <http://www.XYZ.com/WashingMachines/resource/> .

sn:WASH_XYZ
        a                       <http://www.XYZ.com/WashingMachines/ontology#XYZ_Cool> ;
        rdfs:label              "Very cool Washing Machine" ;
        saref:hasFunction       sn:WASH_XYZ-MonitoringFunction , sn:WASH_XYZ-StartStopFunction ;
        saref:hasManufacturer   "XYZ" ;
        saref:hasService        sn:WASH_XYZ-MonitorService , sn:WASH_XYZ-SwitchOnService ;
        saref:hasState          sn:WASHING ;
        s4bldg:isContainedIn    sn:My_Bathroom .

sn:WASH_XYZ-StartStopFunction-OFF_Command     a       saref:OffCommand .

sn:WASH_XYZ-StartStopFunction-Toggle_Command  a       saref:ToggleCommand .

sn:WASH_XYZ-StartStopFunction-ON_Command      a       saref:OnCommand .

sn:WASH_XYZ-MonitoringFunction                a       saref:SensingFunction ;
                saref:hasCommand  sn:WASH_XYZ-MonitoringFunction-WashingMachineStatus .

sn:WASH_XYZ-StartStopFunction                 a       saref:ActuatingFunction ;
                saref:hasCommand  sn:WASH_XYZ-StartStopFunction-Toggle_Command ,
                                  sn:WASH_XYZ-StartStopFunction-OFF_Command ,
                                  sn:WASH_XYZ-StartStopFunction-ON_Command .

sn:WASH_XYZ-MonitoringFunction-WashingMachineStatus a saref:MeteringFunction .

sn:WASH_XYZ-MonitorService                    a       saref:Service .

sn:WASH_XYZ-SwitchOnService                   a       saref:Service .
```

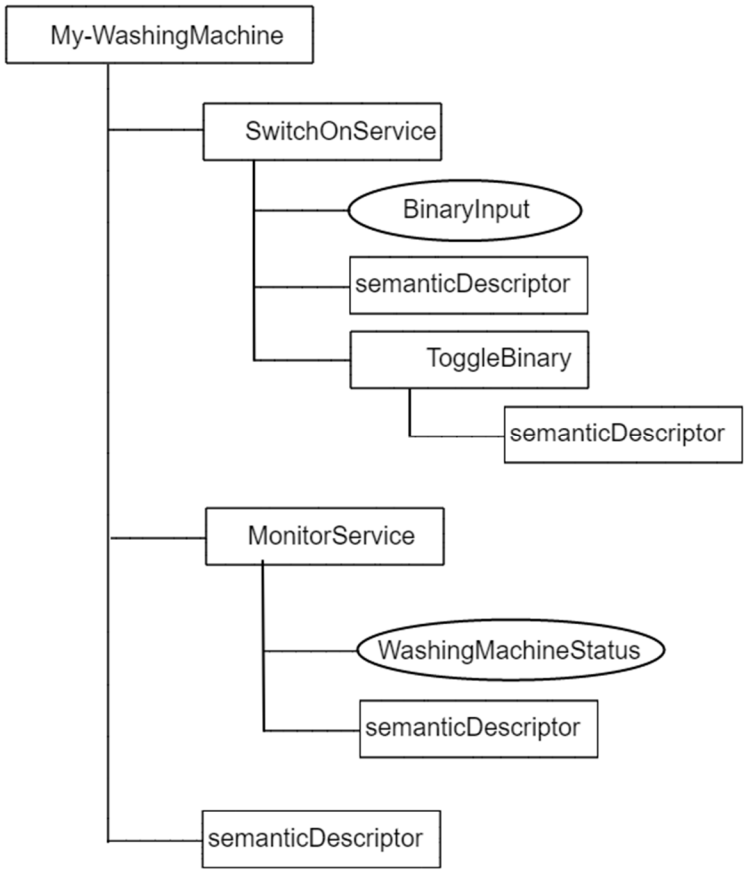

Figure 5: SAREF washing machine model

The SAREF washing machine model inside oneM2M will be composed of:

- two <*flexContainer*> child-resources for Services and their <semanticDescriptor>s are used for modelling the services SwitchOnService and MonitorService;
- the SwitchOnService in turn has a child resource of type <*flexContainer*> for Operations which exposes the Toggle_Command;
- one *customAttribute* of the SwitchOnService <*flexContainer*> is used for holding the values for InputDataPoint: BinaryInput; and
- one *customAttribute* of the MonitorService <*flexContainer*> is used for holding the values for OutputDataPoint: WashingMachineStatus.

7.1.3 Device models using a custom model

oneM2M does not require devices to be modelled using the methods above. The resource tree structure shown here represents a custom model that has an attribute for reading the status of the washing machine and an attribute to set or command the washing machine.

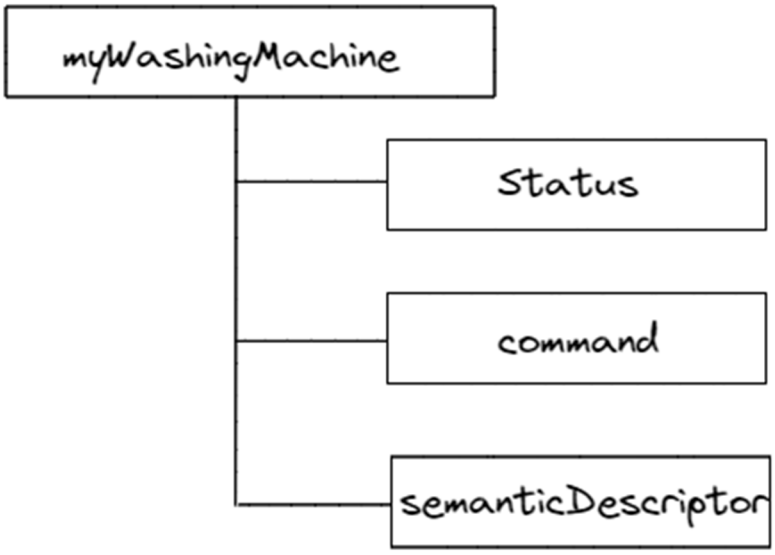

Figure 6: Custom washing machine model

The status is modelled as a <container> resource with <contentInstance> resources that have the following content:

```
{
 "WashingMachineStatus ": "WASHING", // Or "STOPPED", "ERROR"
 "Location": "XYZ",
 "Description": "Very cool Washing Machine"
}
```

Likewise, when setting the state of the device the following payload can be provided:

```
{
 "state": "ON", // Or "OFF", "Toggle"
}
```

7.2 Interworking SAREF and SDT devices

7.2.1 Requirements of interworking

Because the resource tree structure for each of these models is different, clearly the oneM2M primitives needed to access the services of the device are different. However, the goal of interworking device models is to allow a user to issue the same command to perform an operation regardless of which model is used. This can be achieved in a dynamic manner using the oneM2M base ontology to describe each of the services offered by the device and the resources that provide access to those services. For example, the washing machine that has been described offers the following operations:

TURN ON WASHING MACHINE

TURN OFF WASHING MACHINE

TOGGLE THE WASHING MACHINE STATUS

GET STATUS OF WASHING MACHINE

The oneM2M primitives to execute these operations are dependent on the resource tree structure used to model the washing machine. For example, to determine the status of the washing machine for each model the following oneM2M requests and responses are used.

Table 36: Sample operation on washing machine to determine its status

Model	Request	Response
SDT	RETRIEVE /cseBaseName/IPE_ROOT/deviceclothesWasher/runState	{ " currentMachineState ": 3 "machineStates": [1,3,5,6] "currentJobState": 6 "jobStates":[2,3,4,5,6] "progressPercentage":95.0 }
SAREF	RETRIEVE /cseBaseName/IPE_ROOT/My-WashingMachine/MonitorService	{ "WashingMachineStatus":WASHING }
Custom	RETRIEVE /cseBaseName/IPE_ROOT/myWashingMachine/Status/la	{ "WashingMachineStatus":WASHING }

Similarly, to command the washing machine to STOP the following oneM2M primitives are sent.

Table 37: Sample operation on washing machine to stop it

Model	Request
SDT	UPDATE /cseBaseName/IPE_ROOT/deviceClothesWasher/binarySwitch {"state": False }
SAREF	UPDATE /cseBaseName/IPE_ROOT/My-WashingMachine/SwitchOnService {"BinaryInput": False}
Custom	CREATE /cseBaseName/IPE_ROOT/myWashingMachine/Command {"BinaryInput": False}

The SAREF and oneM2M Base Ontologies can be used to describe the services of the washing machine and the oneM2M interface for the services.

7.2.2 Describing the services of a device

Describing the services of the washing machine can be fundamentally the same regardless of which model is used. Especially in the case when describing the same washing machine. The following RDF triples describe the washing machine:

```
sn:WASH_XYZ
        a                       <http://www.XYZ.com/WashingMachines/ontology#XYZ_Cool> ;
        rdfs:label              "Very cool Washing Machine" ;
        saref:hasFunction       sn:WASH_XYZ-MonitoringFunction , sn:WASH_XYZ-StartStopFunction ;
        saref:hasManufacturer   "XYZ" ;
        saref:hasService        sn:WASH_XYZ-MonitorService , sn:WASH_XYZ-SwitchOnService ;
        saref:hasState          sn:WASHING ;
        s4bldg:isContainedIn    sn:My_Bathroom .

sn:WASH_XYZ-StartStopFunction-OFF_Command     a       saref:OffCommand .

sn:WASH_XYZ-StartStopFunction-Toggle_Command  a       saref:ToggleCommand .

sn:WASH_XYZ-StartStopFunction-ON_Command      a       saref:OnCommand .

sn:WASH_XYZ-MonitoringFunction                a       saref:SensingFunction ;
                saref:hasCommand  sn:WASH_XYZ-MonitoringFunction-WashingMachineStatus .

sn:WASH_XYZ-StartStopFunction                 a       saref:ActuatingFunction ;
                saref:hasCommand  sn:WASH_XYZ-StartStopFunction-Toggle_Command ,
                                  sn:WASH_XYZ-StartStopFunction-OFF_Command ,

sn:WASH_XYZ-MonitoringFunction-WashingMachineStatus a saref:MeteringFunction .

sn:WASH_XYZ-MonitorService                    a       saref:Service.

sn:WASH_XYZ-SwitchOnService                   a       saref:Service.
                                sn:WASH_XYZ-StartStopFunction-ON_Command .
```

7.2.3 Describing the interface for using services

7.2.3.1 Describing the interface for the SDT Model

Use of the services described in clause 7.2.2 is different for each model described under clause 7.2.3.

```
sn:WASH_XYZ-SwitchOnService
        a                       oneM2M:Service ;
        oneM2M:hasOperation     sn:WASH_XYZ-SwitchOnServiceOperation .

sn:WASH_XYZ-SwitchOnServiceOperation
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./deviceclothesWasher/binarySwitch"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" .

sn:WASH_XYZ-StartStopFunction-ON_CommandOperation
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./deviceclothesWasher/binarySwitch"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" .

sn:WASH_XYZ-StartStopFunction-OFF_CommandOperation
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./deviceclothesWasher/binarySwitch"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" .

sn:WASH_XYZ-StartStopFunction-TOGGLE_CommandOperation
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./deviceclothesWasher/binarySwitch"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" .

sn:WASH_XYZ-MonitoringFunction-WashingMachineStatusOperation
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./deviceclothesWasher/runState"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "currentMachineState" ;
        oneM2M:oneM2MMethod     "RETRIEVE" .
```

7.2.3.2 Describing the interface for the ontology-based model

```
sn:WASH_XYZ-SwitchOnService
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./My-WashingMachine/SwitchOnService"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "BinaryInput" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" ;

sn:WASH_XYZ-StartStopFunction-ON_Command
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./My-WashingMachine/SwitchOnService"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "BinaryInput" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" ;

sn:WASH_XYZ-StartStopFunction-OFF_Command
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./My-WashingMachine/SwitchOnService"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "BinaryInput" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" ;

sn:WASH_XYZ-StartStopFunction-TOGGLE_Command
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./My-WashingMachine/SwitchOnService/ToggleBinary"^^xsd:anyUri;
        oneM2M:oneM2Mattribute  "" ; // indicates an empty update request see TS-0023
        oneM2M:oneM2MMethod "UPDATE" ;

sn:WASH_XYZ-MonitoringFunction-WashingMachineStatus
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./deviceclothesWasher/runState"^^xsd:anyUri;
        oneM2M:oneM2MMethod     "RETRIEVE" ;
        oneM2M:oneM2Mattribute  "currentMachineState" ;
```

7.2.3.3 Describing the interface for the custom model

```
sn:WASH_XYZ-SwitchOnService
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./myWashingMachine/command"^^xsd:anyUri;
        oneM2M:hasDataRestriction saref:OnCommand, saref:OffCommand, saref:ToggleCommand;
        oneM2M:oneM2MMethod "CREATE" ;

sn:WASH_XYZ-StartStopFunction-ON_Command
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./myWashingMachine/command"^^xsd:anyUri;
        oneM2M:hasDataRestriction saref:OnCommand;
        oneM2M:oneM2MMethod "CREATE" ;

sn:WASH_XYZ-StartStopFunction-OFF_Command
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./myWashingMachine/command"^^xsd:anyUri;
        oneM2M:hasDataRestriction saref:OffCommand;
        oneM2M:oneM2MMethod "CREATE" ;

sn:WASH_XYZ-StartStopFunction-TOGGLE_Command
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./myWashingMachine/command"^^xsd:anyUri;
        oneM2M:hasDataRestriction saref:ToggleCommand;
        oneM2M:oneM2MMethod "CREATE" ;


sn:WASH_XYZ-MonitoringFunction-WashingMachineStatus
        a                       oneM2M:Operation ;
        oneM2M:oneM2MTargetURI "./myWashingMachine/status"^^xsd:anyUri;
        oneM2M:oneM2MMethod "RETRIEVE" ;
```

7.2.4 Discovering Services and Interfaces

By using the oneM2M Base Ontology in the <semanticDescriptor> resources, queries can be sent to the oneM2M CSE to find the services offered by a device and further query those services to discover the oneM2M primitives to access those services.

Here is a list of queries that will be supported for all three of the models:

Query 1: Find all washing machines of manufacturer XYZ

```
SELECT distinct ?wm where {
  ? wm a sn:WASH_XYZ .
}
```

// This should list the IRIs of the washing machines

Query 2: Find all operations offered by washing machines of manufacturer XYZ

```
SELECT distinct ?operation ?wm where {
  ?wm  a   sn:WASH_XYZ  ;
  ?wm oneM2M:hasService ?service ;
  ?service oneM2M:hasOperation ?operation .
}
```

// this should list the IRIs of the operations

Query 3: How do I use operationX of washing machineY

```
DESCRIBE ?operation where {
  ?wm  a   sn:WASH_XYZ  ;
  ?wm  rdfs:label "Very cool Washing Machine" ;
  ?wm oneM2M:hasService ?service ;
  ?service oneM2M:hasOperation ?operation .
}
```

//for washing machine "Very cool Washing Machine" and any of its services, it should provide: oneM2MTargetURI, oneM2MAttribute, oneM2MMethod, and hasDataRestriction_pattern for the CREATE/UPDATE.

7.2.5 Creating <semanticDescriptor> resources

There are two methods for handling semantic queries in oneM2M, described in oneM2M TS-0034 [i.5]. In one method, when the query is received, the oneM2M CSE discovers all the <semanticDescriptor> resources that are within the scope of the query based on authorization policies and the path of the resources. The discovery <semanticDescriptor> resources are used to create an RDF graph from which the query is evaluated. In the second method when the <semanticDescriptor> resources are created they are immediately placed into a triple store.

The ontology-based model shown in clause 7.1.2 shows <semanticDescriptor> resources at multiple levels in the resource tree structure. While the SDT model, and the custom model, show a single <semanticDescriptor> resource at the top level. An examination of the triples in clauses 7.2.2 and 7.2.3 reveal that the information contained in the triples is not dependent on the resource tree hierarchy, in fact these triples can be anywhere in the resource tree structure.

It is suggested that the performance of a query and the complexity of the application creating the <semanticDescriptor> resources can be improved with the following guidelines:

1) Place static information in separate <semanticDescriptor> resources than dynamic information. This is because the SPARQL operations implement UPDATE operations by implementing a DELETE followed by a CREATE operation. Once all information is in a single <semanticDescriptor> resource, the originating entity will handle the extra READ-MODIFY-WRITE functionality. If the dynamic information changes frequently, this can lead to a significant amount of messaging and bandwidth that is not necessary.

    a) The intention of oneM2M is to make data available through data sharing resources such as the <container>, <timeSeries> and <flexContainer>. It is recommended that dynamic information should not be placed in a <semanticDescriptor> resource. Examples of data to not put into an <semanticDescriptor> resource are sensor measurements, current state of a binarySwitch or washingMachine status. Examples of dynamic data that can be put into a <semanticdescriptor> resource is semi-static location of a washing machine.

    b) If placing dynamic data into a <semanticDescriptor> resource is essential to the deployment architecture, then these should be in the <semanticDescriptor> resource by itself, to the possible extent. This will simplify the complexity of the application creating the resource and the discovery and graph store performance.

2) Place all triples for static data in a single <semanticDescriptor> resource, to the possible extent. Some useful criteria for separating the triples into multiple <semanticDescriptor> resources are:

    c) Discovery: it may be desirable to place the triples that describe a service into a separate <semanticDescriptor> resource than the triples that describe the interface and have separate <accessControlPolicies> for these resources. This can support a use case where an application can discover washing machines but requires permission to access the services offered.

    d) Application Specific Inference: it may be desirable to add alignment triples relevant to a specific application or use case.

7.3 Proposed Enhancements Related to the oneM2M Base Ontology

7.3.1 Introduction

There are several observations that are intended to be shared with oneM2M RDM working group for consideration or explanation. The following clauses capture the observations. These will be used to generate the contributions to oneM2M RDM.

7.3.2 Terms to describe the <contentInstance>

The oneM2M Base Ontology [i.2] should be extended to describe a structure of data in the case of <contentInstance> resources being used.

From ETSI TS 118 112 [i.2], the following data property definitions were used to describe the interfaces of the device that was modelled.

hasDataType
hasDataRestriction
oneM2MTargetURI
oneM2MAttribute
oneM2MMethod

For example, using the custom data model for the washing machine, the content of the status container was defined as the following:

```
{
 "WashingMachineStatus ": "WASHING", // Or "STOPPED", "ERROR"
 "Location": "XYZ",
 "Description": "Very cool Washing Machine"
}
```

The oneM2M Base Ontology [i.2] should have a way to describe this. The description could include the format of the data, e.g. json, and the "key" that would contain the desired value(s).

The oneM2M Base Ontology does have definitions of a variety of classes that may be applicable, but it is not clear that they are intended for this purpose, such as: Variable, SimpleTypeVariable, StructuredTypeVariable, and VariableConversion.

7.3.3 SDT Data Type conversions

The oneM2M Base Ontology [i.2] describes classes for VariableConversion that are intended to be used by an IPE to convert values used in oneM2M models, specifically SDT models, to equivalent values that the device uses. So, any device modelled in the oneM2M Service Layer using homogeneous models can be interoperable with an application developed based on that model. This is very good in the case that SDT models are widely adopted.

However, if there are a variety of models used, as described in the present document, and likely most deployments that are candidates for oneM2M, the use of these different models will need to be improved. The methods above demonstrate that controlling devices using different models can be fully achieved using the existing oneM2M Base Ontology. However, it also highlights that applications (as opposed to devices) still have some difficulty. For example, if an application to control washing machine model XYZ uses semantic queries to determine the correct primitives to get the status of the washing machines, the results can be seen to differ in such a way as to require the application to have some specific domain knowledge.

The oneM2M Base Ontology [i.2] should be extended such that an application can indicate how the response should be returned. For example, the washing machine status returned in the models above could be "WASHING" for the ontology-based model and the custom model. But for the SDT-based model the equivalent value is "3". Applications cannot handle this type of difference.

Also related to SDT models is a way to apply an SDT interface onto a device modelled differently. Take for example a smartphone application that is developed to control washing machines using the SDT model. If that application issues semantic queries to discover washing machines and gets the results described above, can there be an interworking from the application side such that a primitive using the SDT model-based command can be effectively converted to the other type of command. This can leverage the VariableConversion capabilities in the Base Ontology.

7.3.4 oneM2M <group> of heterogeneous devices

The oneM2M group functionality supports homogenous resources and heterogenous resources. Conceptually it would be possible to have a group of washing machines or a group of get washing machine status or a group for turn all washing machines OFF. However, the group fanout procedures will not work as expected. This extends the discussion from clause 7.3.3 where a primitive based on a SDT model to turn on the washing machine will not work for the other models.

Perhaps there is a way to issue a "Semantic primitive" where the primitive indicates a saref:OffComand should be sent to all of the devices, which would then be processed by all of the washing machines in the group that offer that service.

7.3.5 oneM2M Base Ontology examples

The oneM2M Base Ontology [i.2] contains a large collection of classes to describe devices and their capabilities and features. The ETSI TS 118 112 [i.2] specification would benefit by adding concrete examples to each of the classes to demonstrate how they would work in a oneM2M deployment, including the types of queries that are enabled by these classes. This may lead to a reduction in the vocabulary needed or the identification of new classes that would be beneficial to the oneM2M community.

The description of classes and properties should include specific examples of how to use the base ontology to describe devices/services in oneM2M using specific resources as examples, including at a minimum <container>/<contentInstance> and <flexContainer> (This could also apply to the labels attribute. In general to anything that is supposed to contain application-specific data model information).

History

Document history		
V1.1.1	May 2022	Publication