SAREF4EE Documentation

Laura Daniele, TNO *version0.1, 28-12-2015*

1. Introduction

SAREF4EE is the extension of SAREF for the EEbus-E@h project. In the ontology documentation available online[1] and in this document we distinguish between SAREF and SAREF4EE using the prefixes `saref:` and `s4ee:`, respectively.

In order to demonstrate product interoperability and the capability to fulfill Demand Response scenarios, SAREF4EE represents the information exchanged in various use cases[2] commonly defined by the Energy@Home and EEbus associations. The information represented in SAREF4EE can be categorized as follows:

1. Configuration information exchanged in the use case "Remote Network Management" between devices that want to connect to each other. For example, a new dishwasher that wants to register to a Customer Energy Manager (CEM) in the cloud or on a gateway, as it is shown in Figure 1. The source used for defining the configuration information in SAREF4EE is the *EEBus Technical Report, Protocol Specification- Remote Network Management, version 1.0.0.2, 2015-09-19*.

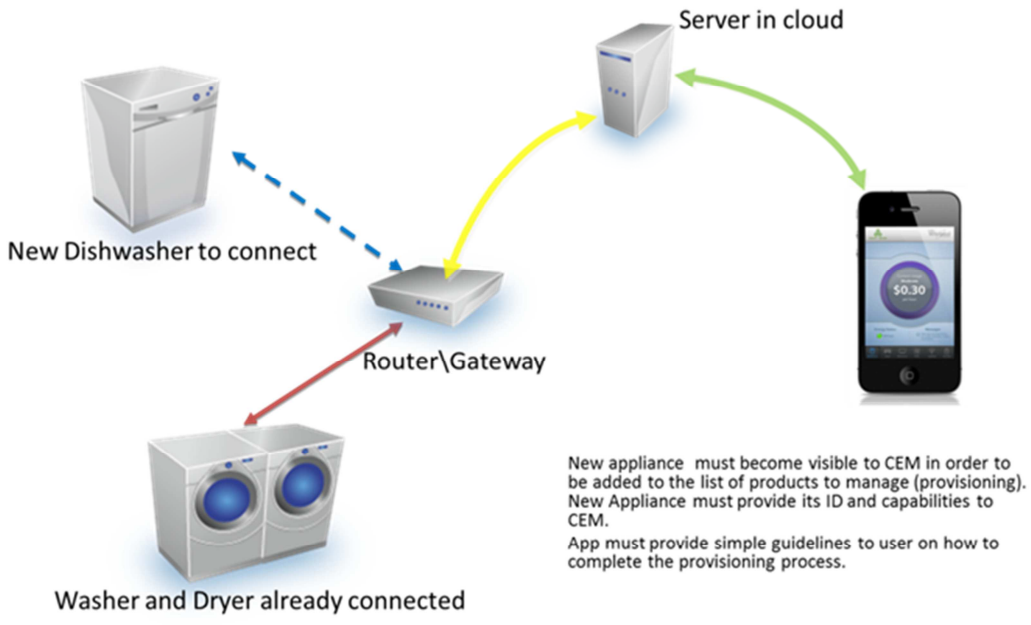

Figure 1- Remote Network Management: new appliance registration

[1] Temporarily available at http://ontology.tno.nl/saref4ee, it may be moved to an Energy@Home or EEbus server later on

[2] Assumptions for these use cases are that i) Multi-tariffs and Incentives to consume\curtail are available; ii) Information from smart meter is available On Demand; iii) User UI will be on Smart phone\Tablet; iv) CEM can be either in cloud or on a gateway.

2. Schedule information exchanged in the use cases "Appliance scheduling through CEM and remote start" (see Figure 2.a) and "Automatic cycle rescheduling" (see Figure 2.b). The source used for defining the schedule information in SAREF4EE is the *General Message Structures, version 0.1.1, 2015-10-07*.

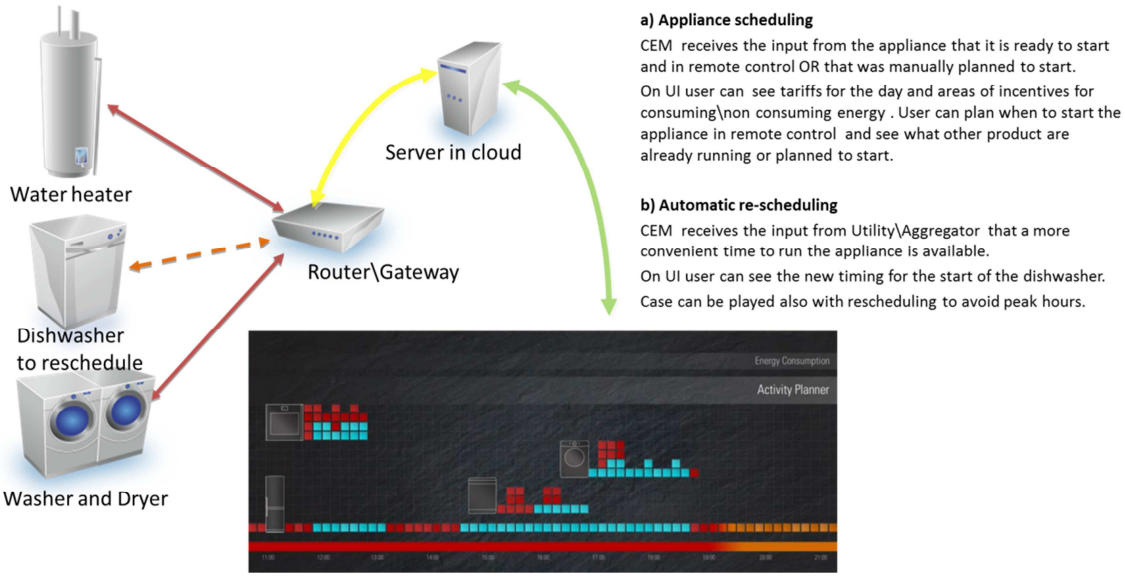

Figure 2- Appliance (re)scheduling

3. Monitoring and control information exchanged in the "Communicate appliance status and information on manually planned cycles" use case shown in Figure 3. The source used for defining the monitoring and control information in SAREF4EE is the *Energy@Home Data Model version 1.0*.

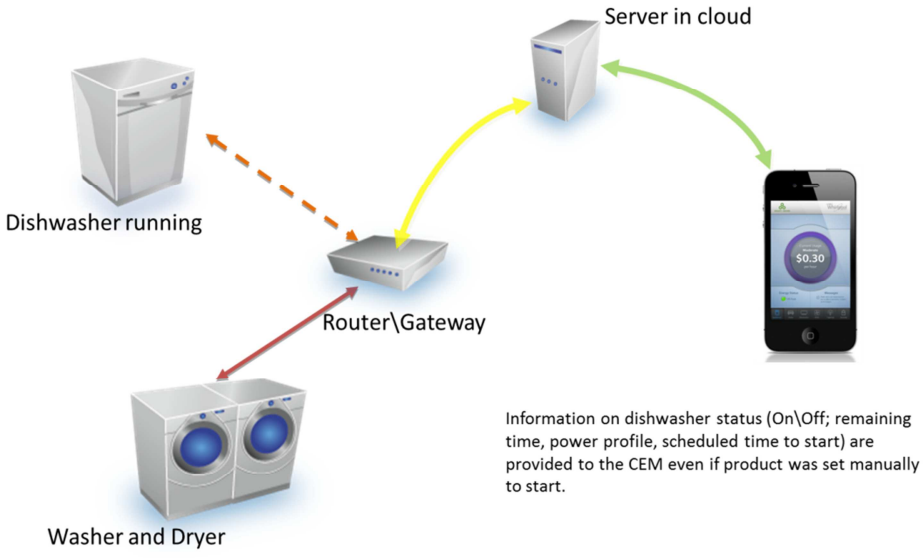

Figure 3- Appliance status and information on manually planned cycles

4. Event-based information exchanged in the use case "Demand Response" to model demand response events such as, for example, the direct load management and power curtailing shown

in Figure 4. The source used for defining the event based information in SAREF4EE is the *General Message Structures, version 0.1.1, 2015-10-07* specification[3].

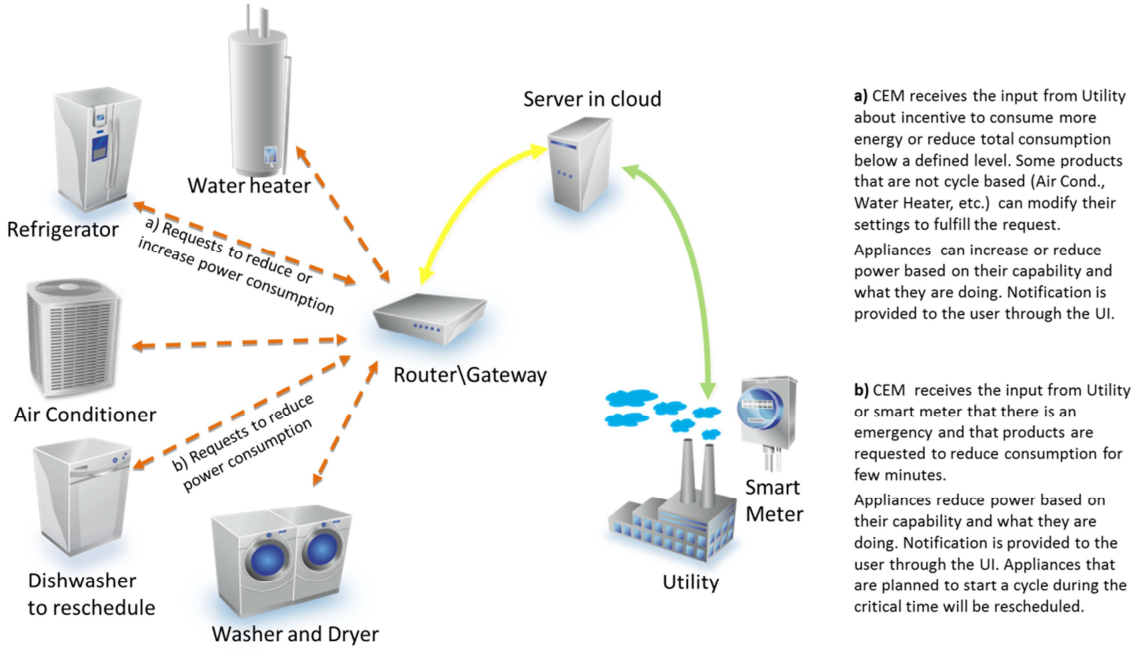

Figure 4- Direct load management and power curtailing

The next sections describe the classes and properties of SAREF4EE. For the sake of readability, we structured the presentation according to the different types of information mentioned above, namely configuration information, schedule information, monitor and control information, and event-based information.

2. Configuration Information

This section addresses the use case "Remote Network Management" that describes how devices can exchange configuration information on their mutual functionality in order to connect to each other. The classes of interest are `s4ee:Device`, `s4ee:Address`, `s4ee:DeviceConnection`, `s4ee:DeviceConnectionSetup`, `s4ee:NativeSetup`, `s4ee:CandidateSetup`, `s4ee:ScanSetup` and `s4ee:JoinModeConfiguration`, as shown in Figure 5.

A `s4ee:Device` is a subclass of a `saref:Device`, i.e., it inherits the properties of the more general `saref:Device` extending it with additional properties that are specific for SAREF4EE. When connecting to another device, such as the CEM, a `s4ee:Device` can be associated to zero or more node addresses using the `s4ee:hasNodeAddress` property. The details of the `s4ee:Address` class are out of the scope of SAREF4EE since they depend on specific implementations technologies. Figure 5 shows that we indicate two possible types of addresses as examples, i.e., `s4ee:IPaddress` and `s4ee:MACaddress`, but any other type of address can be added to accommodate different needs.

[3] Note that it is emphasized there that the one described is only a temporary specification: “it is expected a future COSEM specification to replace it, since COSEM has been chosen for everything coming from the Smart Meter side (such as demand response)”.

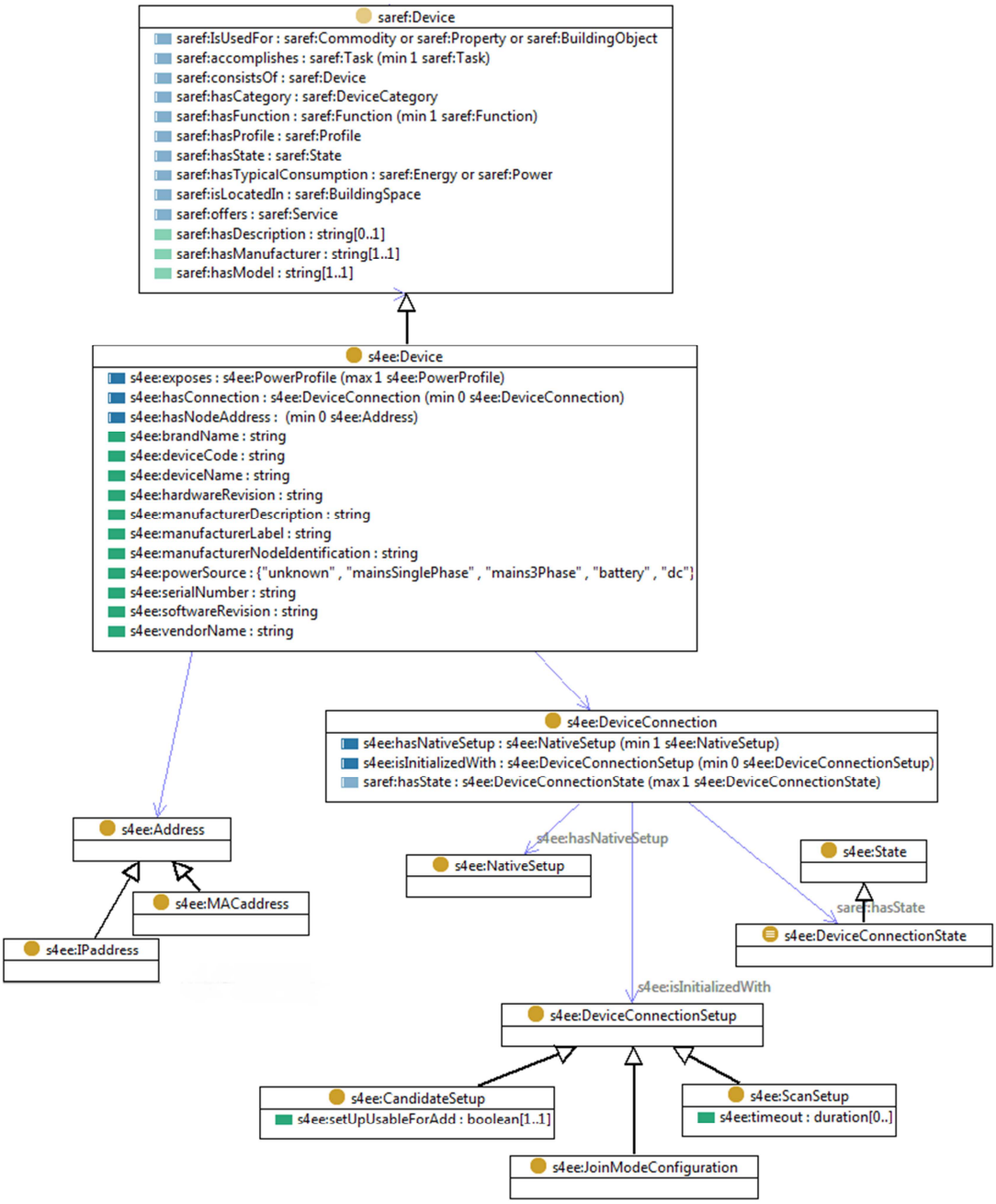

Figure 5 – s4ee:DeviceConnection, s4ee:DeviceConnectionSetup, s4ee:Address classes and their properties

The `s4ee:DeviceConnection` class models the connection of a device with another device in the network. In particular, a `s4ee:Device` can have zero or more connections at the same time (`s4ee:hasConnection` property). Before the connection is established there is an initialization that may consists of zero or more setup phases (`s4ee:DeviceConnection s4ee:isInitializedWith min 0 s4ee:DeviceConnectionSetup`). These setup phases are subclasses of the `s4ee:DeviceConnectionSetup` class and can be of the following 3 types:

- `s4ee:CandidateSetup` which represents the information necessary to initially connect to the device;

- `s4ee:ScanSetup` which represents the information about which other devices are accessible in the network; and
- `s4ee:JoinModeConfiguration` which represents alternative information to scan setup for configuring the default or "background" behaviour of the native communications technology implementation with regards to the announcement or acceptance of other devices.

Once a device connection is established, it is then characterized by a `s4ee:Native Setup`, which specifies the information about the connection with a certain device. Finally, the device connection is characterized by a state (optional) specified by the `s4ee:DeviceConnectionState` class, which is a subclass of the `s4ee:State` class and can assume one of the following values: "`added`", "`failed`", "`modified`", "`removed`" or "`succeeded`".

3. Schedule information

The classes of interest for the "Appliance scheduling through CEM and remote start" and "Automatic cycle rescheduling" use cases are `s4ee:Device`, `s4ee:PowerProfile`, `s4ee:Alternative`, `s4ee:PowerSequence` and `s4ee:Slot`, which are shown in Figure 6.

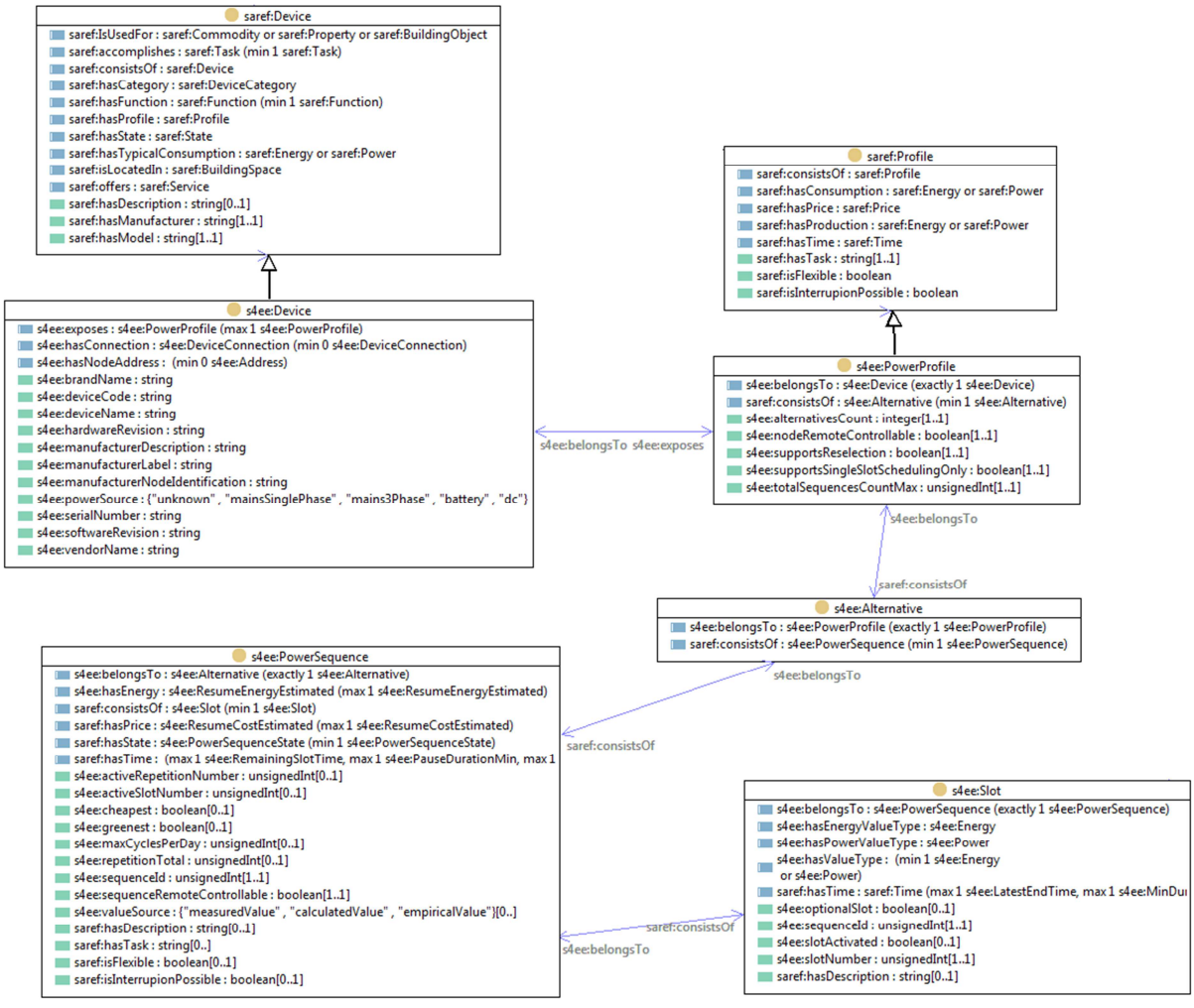

Figure 6 - s4ee:Device, s4ee:PowerProfile, s4ee:Alternative, s4ee:PowerSequence, s4ee:Slot classes and their properties

A s4ee:PowerProfile is a subclass of a saref:Profile, i.e., it inherits the properties of the more general saref:Profile extending it with additional properties that are specific for SAREF4EE. The s4ee:PowerProfile is used by a s4ee:Device to expose the power sequences that are potentially relevant for the CEM. A s4ee:Device can expose at most one s4ee:PowerProfile, which consists of one or more alternative plans (s4ee:Alternative class). A s4ee:Alternative consists of one or more power sequences (s4ee:PowerSequence class), and a s4ee:PowerSequence consists of one or more slots (s4ee:Slot class). Inversely, a s4ee:Slot belongs to only and exactly one s4ee:PowerSequence, which, in turn, belongs to only and exactly one s4ee:Alternative, which, in turn, belongs to only and exactly one s4ee:PowerProfile. A s4ee:PowerProfile belongs to only and exactly one s4ee:Device.

Table 1 summarizes the properties that characterize a s4ee:PowerProfile.

Property	**Definition**
s4ee:alternativesCount **exactly** 1 xsd:integer	Number of alternatives provided by a power profile.
s4ee:nodeRemoteControllable **exactly** 1 xsd:Boolean	Whether the device is configured for remote control by the CEM. This refers to the selection chosen by the user on the remote control feature of the device.
s4ee:supportsReselection **exactly** 1 xsd:boolean	Whether the device restricts the number of sequence re-selections by the CEM. If set to TRUE, there is no restriction, i.e., within a given alternative the CEM may first choose one sequence, alter the selection by configuring another sequence later on, then alter the selection again, etc. If set to FALSE, the device permits the CEM to select a sequence of an alternative only one time.
s4ee:supportsSingleSlotSchedulingOnly **exactly** 1 xsd:boolean	Whether the device permits the modification of more than one slot per configuration command. If set to TRUE the device does NOT permit this modification.
s4ee:totalSequencesCountMax **exactly** 1 xsd:unsignedInt	Total number of sequences supported by the device, i.e., the sum of all power sequences across all alternatives.

Table 1 - Properties of the s4ee:PowerProfile class

Table 2 summarizes the properties that characterize a s4ee:PowerSequence.

Property	**Definition**
s4ee:sequenceId **exactly** 1 xsd:unsignedInt	An endpoint-wide unique sequence identifier.
saref:hasDescription **max** 1 xsd:string	Textual description for the power sequence.
saref:isFlexible **max** 1 xsd:Boolean *(called "isStoppable" in EEbus)*	If the power sequence is stoppable by the CEM, this element is TRUE. Otherwise it SHALL be omitted.
saref:isInterrupionPossible **max** 1 xsd:Boolean *(called "isPausable" in EEbus)*	If the power sequence is pausable by the CEM, this element is TRUE. Otherwise it SHALL be omitted.
saref:hasTask **min** 0 xsd:string *(called "taskIdentifier" in EEbus)*	Used by a device that wants to uniquely identify reoccurring types of power sequences. For example, specific types of washing cycles with specific parameters SHOULD have the same saref:hasTask value every time they are offered using power sequences.
s4ee:activeRepetitionNumber **max** 1 xsd:unsignedInt	The current repetition of the sequence of slots. SHALL be present if s4ee:repetitionsTotal is present and has a value > 1. Otherwise, it SHALL be absent.
s4ee:activeSlotNumber **max** 1 xsd:unsignedInt	If s4ee:PowerSequenceState is set to "running" or "paused" this element SHALL contain the currently active slot. Otherwise it SHALL be omitted.
s4ee:cheapest **max** 1 xsd:boolean	If present and set to TRUE, the CEM shall try to apply a

	configuration that minimises the user's energy bill for this power sequence. Absence of this element is equal to the presence with value FALSE.
s4ee:greenest **max** 1 xsd:boolean	If present and set to TRUE, the CEM shall try to optimise the configuration towards the maximum availability of renewable energy. Absence of this element is equal to the presence with value FALSE.
s4ee:maxCyclesPerDay **max** 1 xsd:unsignedInt	The maximum amount of starts that the device allows per day.
s4ee:repetitionTotal **max** 1 xsd:unsignedInt	If a power sequence repeats its sequence of slots, the element MUST be present and contains the total number of repetitions. Absence of the element is equal to a presence with a value of 0 (zero). SHALL be absent if the value is 1.
s4ee:sequenceRemoteControllable **exactly** 1 xsd:boolean	Whether the sequence is modifiable (if value is TRUE) or not (if value is FALSE). Modifiability is required to configure power sequences and slots. It is also required to change a power sequence state.
s4ee:valueSource **min** 0 {"measuredValue", "calculatedValue", "empiricalValue"}	The source (origin/foundation) of the measurement forecasted values for this power sequence. If absent, the source is undefined.
s4ee:hasEnergy **max** 1 s4ee:ResumeEnergyEstimated	The additional energy the device will consume before resuming its normal operation (after a pause). This is only an estimated value which will not be added to the value stated in any slot value information.
saref:hasPrice **max** 1 s4ee:ResumeCostEstimated	The additional costs for the resumption of a device to its normal operation (after a pause).
saref:hasState **min** 1 s4ee:PowerSequenceState	The current state of the power sequence. It can assume one of the following values: 'running', 'paused', 'scheduled', 'scheduled paused', 'pending', 'inactive', 'completed', or 'invalid'.
saref:hasTime **max** 1 s4ee:ActiveDurationMax	The active maximum duration the power sequence can run without interruption.
saref:hasTime **max** 1 s4ee:ActiveDurationMin	The active minimum duration the power sequence can run without interruption.
saref:hasTime **max** 1 s4ee:ActiveDurationSumMax	The active maximum duration the power sequence can run in total (summation of all active times).
saref:hasTime **max** 1 s4ee:ActiveDurationSumMin	The active minimum duration the power sequence must run in total (summation of all active times).
saref:hasTime **min** 1 s4ee:StartTime	The start time of the power sequence. SHALL be present.
saref:hasTime **max** 1 s4ee:EarliestStartTime	SHALL state the earliest possible start time for the whole power sequence.
saref:hasTime **max** 1 s4ee:EndTime	The end time of the power sequence. If the value is available, it SHALL be denoted here. Otherwise the element SHALL be omitted.
saref:hasTime **max** 1 s4ee:LatestEndTime	The latest possible end time for the whole power sequence.
saref:hasTime **max** 1 s4ee:ElapsedSlotTime	If the power sequence state is set to 'running' or 'paused' AND the slot is determined, this element CAN contain the time the slot has already been in 'running' state (this also means the value remains constant during a 'paused' state). Otherwise it SHALL be omitted.
saref:hasTime **max** 1 s4ee:RemainingSlotTime	If the power sequence state is set to 'running' or 'paused' AND the slot is determined, this element SHALL contain the time the slot still needs to be in 'running' state (this also means the value remains constant during a 'paused' state). Otherwise it SHALL be omitted.

saref:hasTime **max** 1 s4ee:PauseDurationMax	The maximum duration the power sequence can pause after the end of an activity.
saref:hasTime **max** 1 s4ee:PauseDurationMin	The minimum duration the power sequence can pause after the end of an activity.

Table 2 - Properties of the s4ee:PowerSequence class

Table 3 summarizes the properties that characterize a `s4ee:Slot`.

Property	**Definition**
s4ee:slotNumber **exactly** 1	A power sequence Id-wide unique slot identifier.
saref:hasDescription **max** 1	Textual description for the slot.
s4ee:sequenceId **exactly** 1	Endpoint-wide unique identifier of the power sequence to which the slot belongs to.
s4ee:optionalSlot **max** 1	It is set to TRUE if the slot can be omitted, otherwise the element SHALL be omitted or set to FALSE. Note: this element applies to every repetition of the slot number.
s4ee:slotActivated **max** 1	If the slot is optional, i.e. s4ee:optionalSlot is set to TRUE, this element reflects the current status of the slot (TRUE = the slot will be executed, FALSE = the slot will not be executed). If the slot is not optional, this element SHALL be absent.
s4ee:hasValueType **min** 1 (s4ee:Energy **or** s4ee:Power)	The type of energy or power (subclasses of saref:Energy and saref:Power). The energy can be of type s4ee:EnergyMin, s4ee:EnergyMax, s4ee:EnergyExpected, s4ee:EnergyStandardDeviation or s4ee:EnergySkewness. The power can be of type s4ee:PowerMin, s4ee:PowerMax, s4ee:PowerExpected, s4ee:PowerStandardDeviation or s4ee: Power Skewness.
saref:hasTime **max** 1 s4ee:DefaultDuration	The duration of the slot (in case of 'determined slot'). If the slot has a configurable length, this element SHALL reflect the currently configured length.
saref:hasTime **max** 1 s4ee:Duration	The duration of the slot (if the slot has a configurable length). Otherwise it CAN state the fixed duration of the slot.
saref:hasTime **max** 1 s4ee:MaxDuration	The maximum supported configuration (if the slot has a configurable duration). Note: this element applies to the first repetition of the slot number only.
saref:hasTime **max** 1 s4ee:MinDuration	The minimum supported configuration (if the slot has a configurable duration). Note: this element applies to the first repetition of the slot number only.
saref:hasTime **max** 1 s4ee:DurationUncertainty	The uncertainty of the duration given in the s4ee:Duration class.
saref:hasTime **max** 1 s4ee:StartTime	The start time of the slot. SHALL be present.
saref:hasTime **max** 1 s4ee:EarliestStartTime	SHALL state the earliest possible start time for the slot.
saref:hasTime **max** 1 s4ee:EndTime	The end time of the slot. The following equation SHALL apply: EndTime - StartTime = DefaultDuration.
saref:hasTime **max** 1 s4ee:LatestEndTime	The latest possible end time for the slot.
saref:hasTime **max** 1 s4ee:RemainingPauseTime	The duration that the current slot permits being paused. This element SHALL ONLY be present if the power sequence is interruptible (pausable), i.e., saref:isInterrupionPossible has value TRUE.

Table 3 - Properties of the s4ee:Slot class

Figure 7 shows a sample instance of a power profile, called `s4ee:PowerProfile_Device1`, which belongs to an instance of a device called `s4ee:Device_1` and consists of two possible alternatives. In particular, the first alternative, called

s4ee:Alternative1_PowerProfile_Device1, consists of a power sequence called s4ee:PowerSequence_PS123456 that has a sequence ID with value 123456. This power sequence consists of a slot called s4ee:Slot1_PS123456.

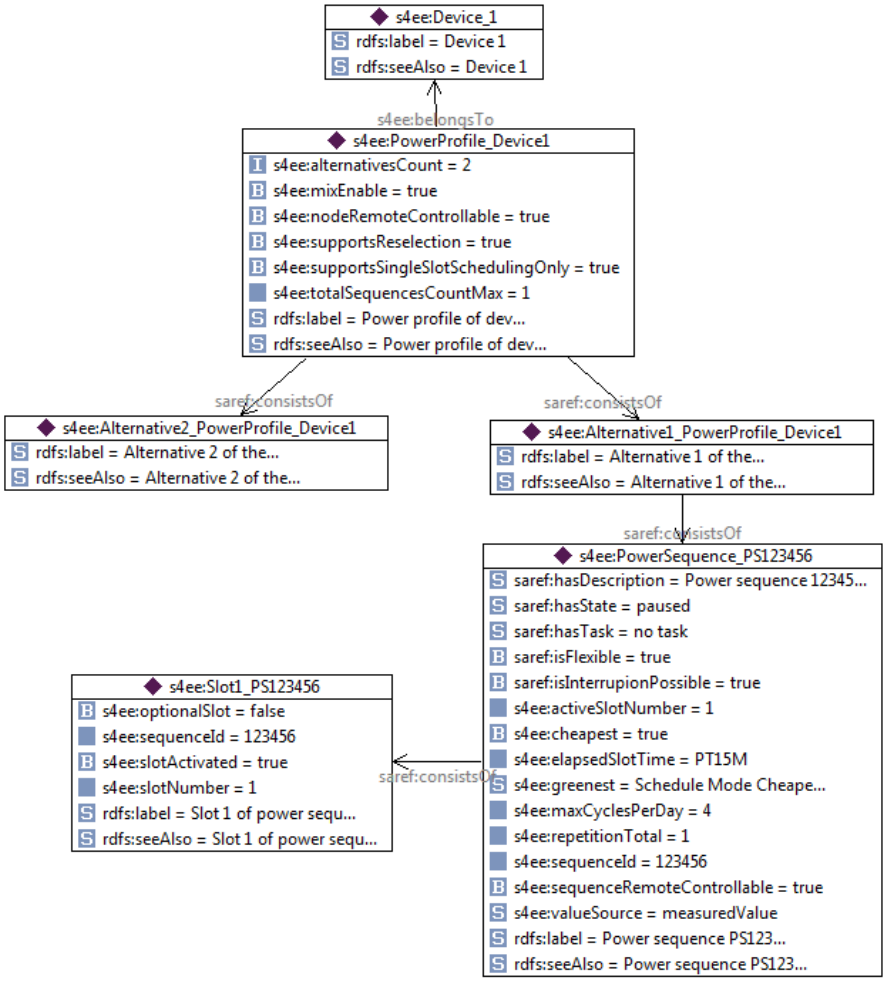

Figure 7 – Instance of a s4ee:PowerProfile

4. Monitoring and Control information

Appliances are connected devices for which it is possible to identify the type of load (e.g., type of appliance, supplier name, firmware version, etc.) and to monitor and control the start and the status of operation, to communicate information to diagnose problems, as well as the transmission of statistical information and the tunneling of manufacturer proprietary information of the appliance. The classes of interest to represent the monitor and control information of SAREF4EE are s4ee:Appliance, s4ee:ApplianceParameter, s4ee:ApplianceParameterTable, s4ee:ParameterTablePoint, s4ee:Value, s4ee:ApplianceParameterState, s4ee:ApplianceWorkingMode, s4ee:ApplianceParameterSet, s4ee:ApplianceParameterSettings, s4ee:Expression, s4ee:ApplianceParameterCompatibilityAction, s4ee:ApplianceControl and s4ee:ApplianceMonitor.

A s4ee:Appliance is as a specialization of a s4ee:Device and therefore also a specialization of a saref:Device, as shown in Figure 8.

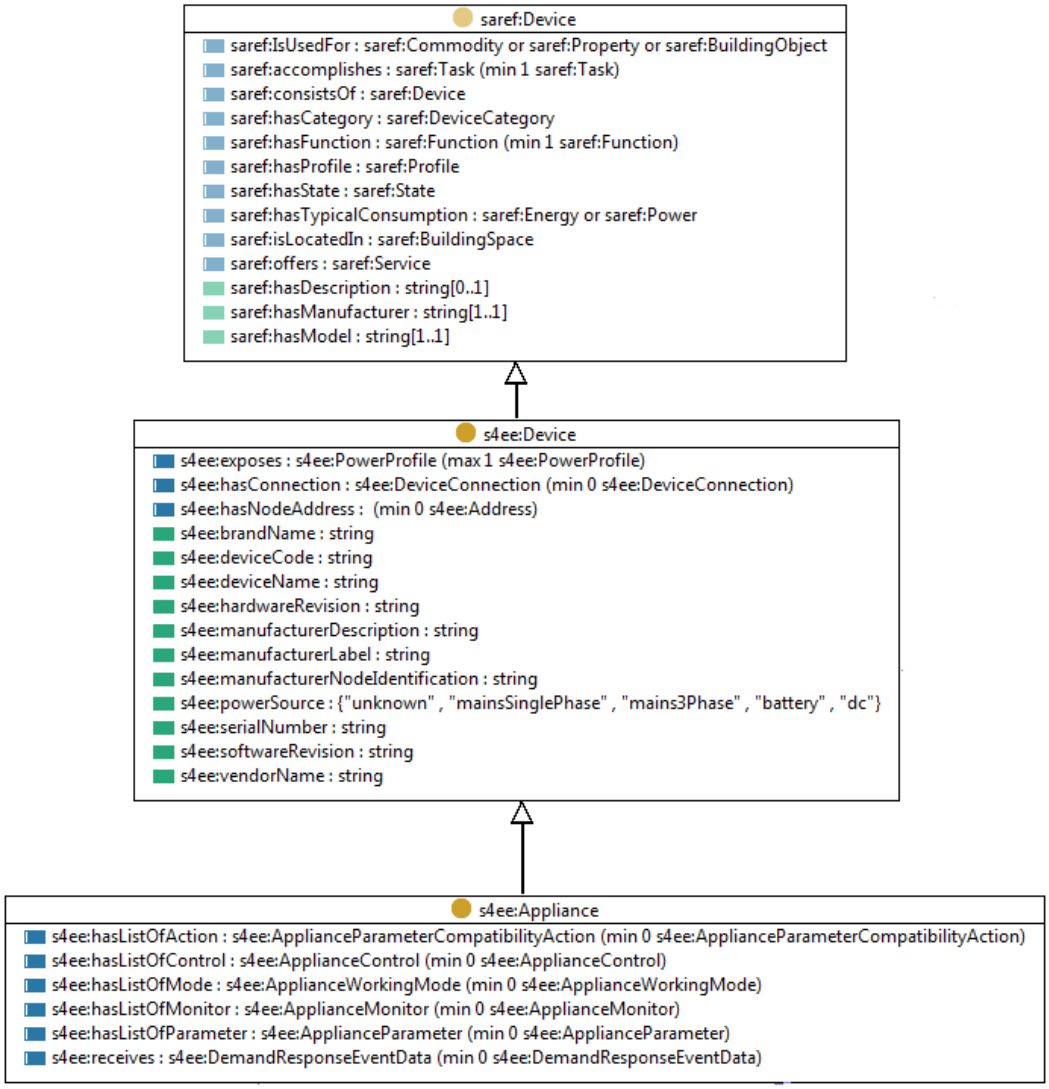

Figure 8 – The s4ee:Appliance class and its properties

A s4ee:Appliance is linked to parameters , available working modes, controls and measurements, as follows:

– it has a list of zero or more parameters (s4ee:ApplianceParameter class in Figure 9), each representing a particular function mode such as "Temperature", "Spin", "Prewash" or "Iron Min";
  - Each s4ee:ApplianceParameter is described by a s4ee:ParameterTable, which can be of type s4ee:StepParameterTable, s4ee:PointwiseParameterTable, s4ee:BooleanParameterTable or s4ee:DateParameterTable. All these tables define the type of permission for a certain parameter (i.e., "read only", "write only" or "read and write") and its unit of measure (saref:isMeasuredIn property). The s4ee:StepParameterTable is additionally characterized by at least one minimum value, number of set points and steps. The

s4ee:PointwiseParameterTable is characterized by a point with one or more values (s4ee:hasPoint min 1 property) described by the s4ee:ParameterTablePoint class.

- Each s4ee:ApplianceParameter is associated to a state (saref:hasState exactly 1 s4ee:ApplianceParameterState) which can be used to represent the actual parameter values by means of the s4ee:ApplianceMonitor class, and to set new values by means of the s4ee:ApplianceControl class.

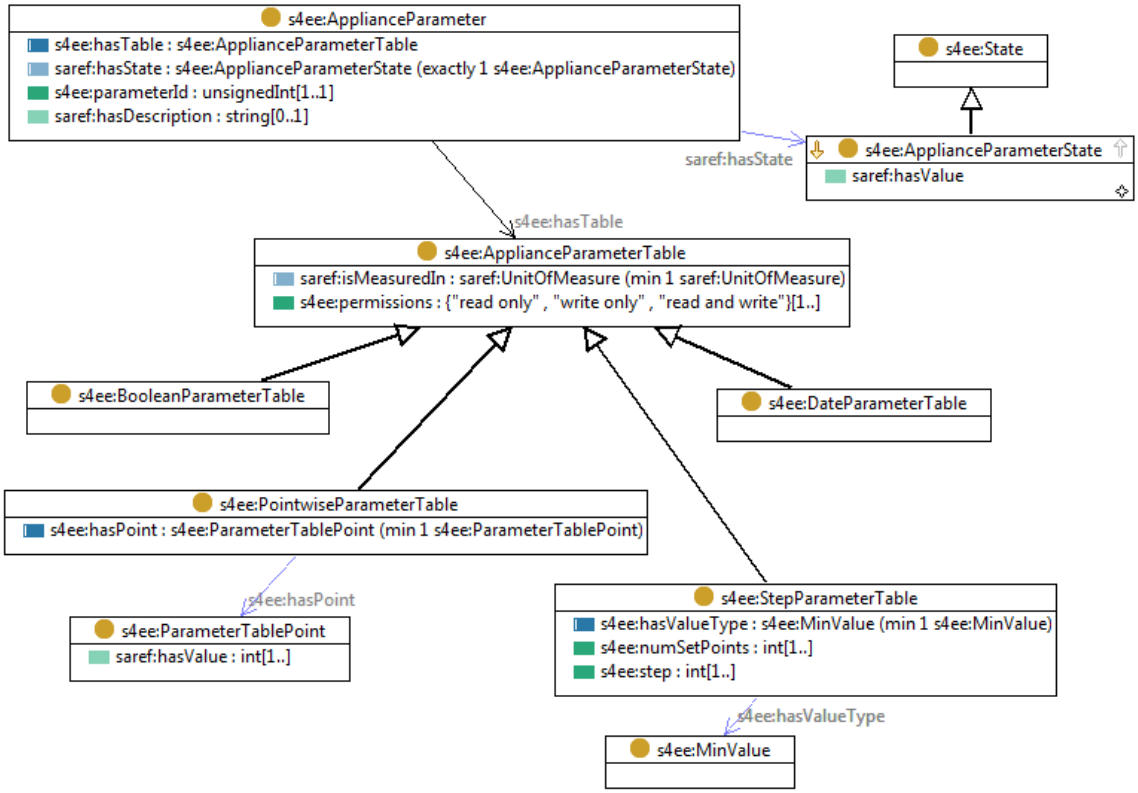

Figure 9 – The s4ee:ApplianceParameter class and its properties

- It has a list of zero or more working modes (s4ee:ApplianceWorkingMode class in Figure 10) each representing a particular working mode such as "Synthetics", "Mix 30" or "Super Cool";
  - A working mode has an ID (s4ee:workingModeId exactly 1 property), a name (saref:hasName exactly 1 property) and a list of zero or more sets (s4ee:ApplianceParameterSet class) representing the sets of enabled parameters for that working mode. A s4ee:ApplianceParameterSet can have zero or more settings (s4ee:ApplianceParameterSettings class)and is selected according to certain conditions defined in the s4ee:Expression class. The set "0" is the default set and is selected when no condition is true.
    - The s4ee:ApplianceParameterSettings class is characterized by the parameter ID (s4ee:parameterId exactly 1 property) and a number of values for that parameter that are subclass of the s4ee:Value class, i.e., s4ee:AvoidedValue (list of not admitted values), s4ee:DefaultValue (default value of the parameter), s4ee:MaxValue (maximum value that the

parameter could be set) and `s4ee:MinValue` classes (minimum value that the parameter could be set). The `s4ee:ApplianceParameterSettings` class also has a boolean property to specify whether the settings under consideration are active or not (`s4ee:isActive` property).

- The `s4ee:Expression` class is characterized by a value (`s4ee:hasValueType exactly 1` property), the parameter ID (`s4ee:parameterId exactly 1` property) that identifies the parameter whose current set point has to be compared, a math operator (`s4ee:mathOperator` exactly 1 property) such as "==" , "!=" , ">" , "<" to define set points equal, different, above or below the expression value, and logical connectives (`s4ee:logicalConnective min 0` property) such as "AND" and "OR" that could be used to connect different expressions.

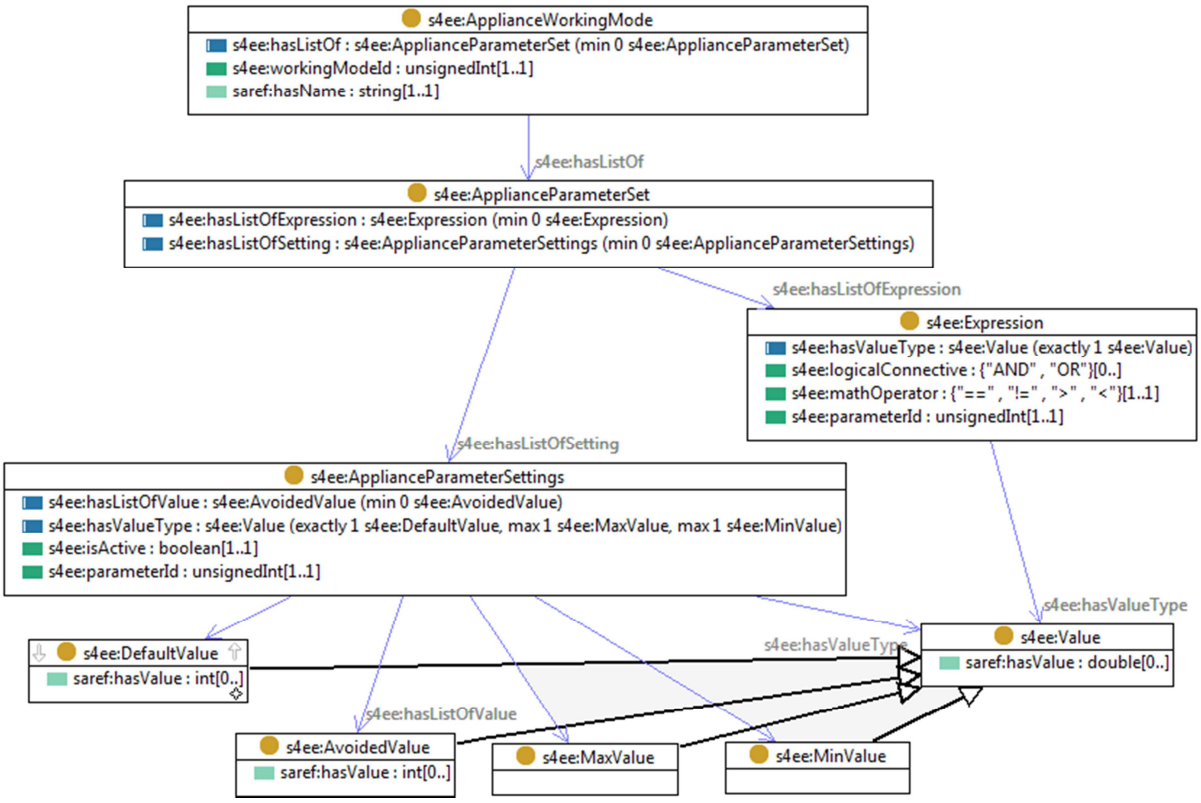

Figure 10 – The s4ee:ApplianceWorkingMode class and its properties

– It has a list of zero or more actions to be executed in case of incompatibility with other parameters (`s4ee:ApplianceParameterCompatibilityAction` class in Figure 11).

  - The `s4ee:ApplianceParameterCompatibilityAction` class specifies incompatible parameters (`s4ee:hasAffectedParameter min 1` property), and has at least one expression (`s4ee:hasExpression min 1` property). If this expression turns TRUE, then one of the following types of actions will be executed:
    - `s4ee:action_1_reset_to_OFF_value` (it shall be reset)
    - `s4ee:action_2_disabled` (it shall be disabled)
    - `s4ee:action_3_set_to_MaxValue` (it shall be set to maximum value)

- `s4ee:action_4_set_to_default_value` (it shall be set to default value).

The property s4ee:`hasValue min 0 s4ee:MaxValue` expresses the maximum value to be used in case of `s4ee:action_3_set_to_MaxValue`.

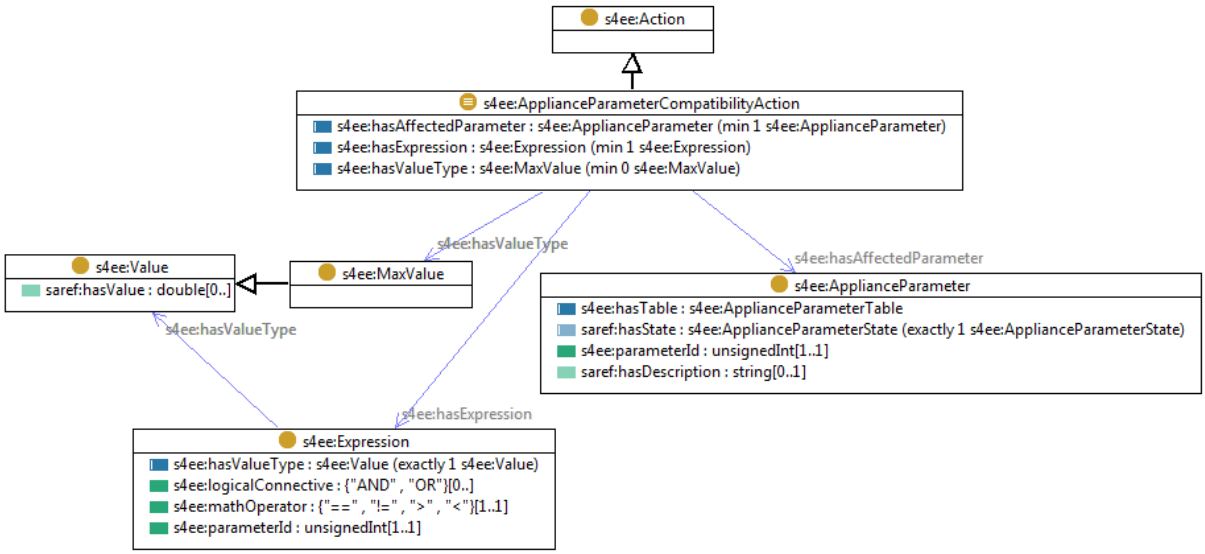

Figure 11 – The s4ee:ApplianceCompatibilityAction class and its properties

– It has a list of zero or more measurements that represent the actual parameter values for the appliance (`s4ee:ApplianceMonitor` class in Figure 12). These measurements can be sent by the appliance automatically as a status notification, or after a specific request from the CEM. The notification contains the information related to the current state of the appliance, i.e., parameter ID, its current value and, optionally, the maximum and minimum values that the parameter can assume.

– It has a list of zero or more control actions (`s4ee:ApplianceControl` class in Figure 12), such as command actuation or the setting of working modes and parameters, to control zero or more states of the appliance (`s4ee:ApplianceParameterState` class). The `s4ee:ApplianceControl` class also has a boolean property to specify whether the controls under consideration are active or not (`s4ee:isActive` property).

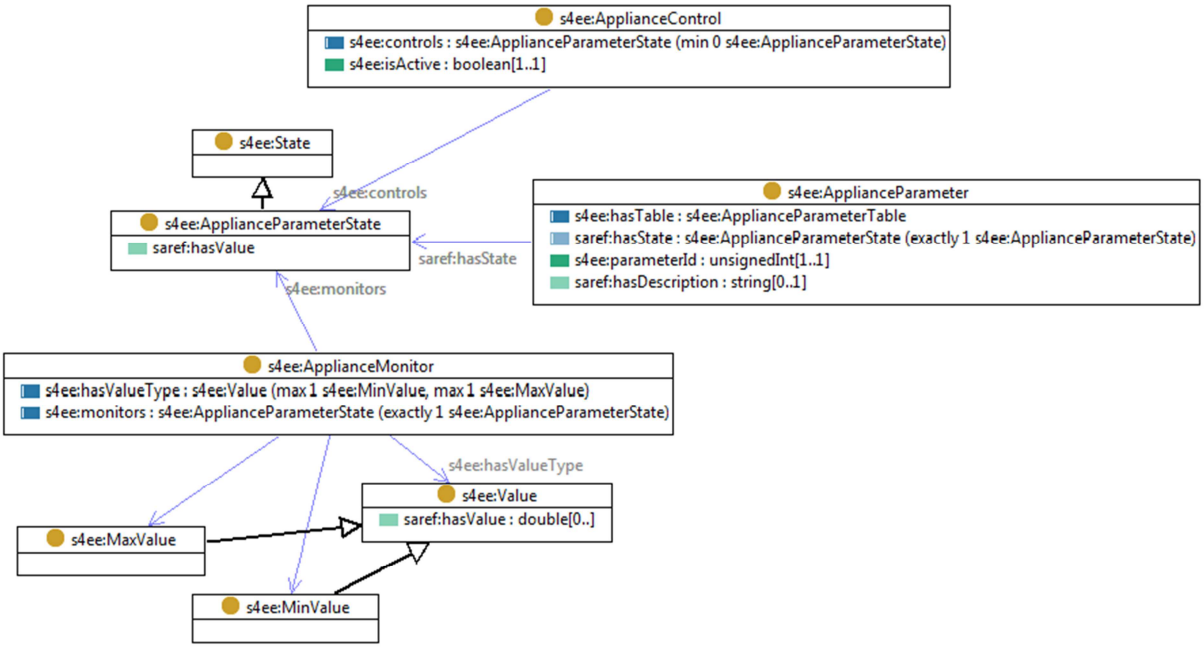

Figure 12 – The s4ee:ApplianceMonitor and s4ee:ApplianceControl classes and their properties

5. Event-based information

The classes of interest to model demand response events are `s4ee:DemandResponseEventData`, `s4ee:DemandResponseEventAction`, `s4ee:DemandResponseStateData` and `s4ee:DemandResponseState`, as shown in Figure 13.

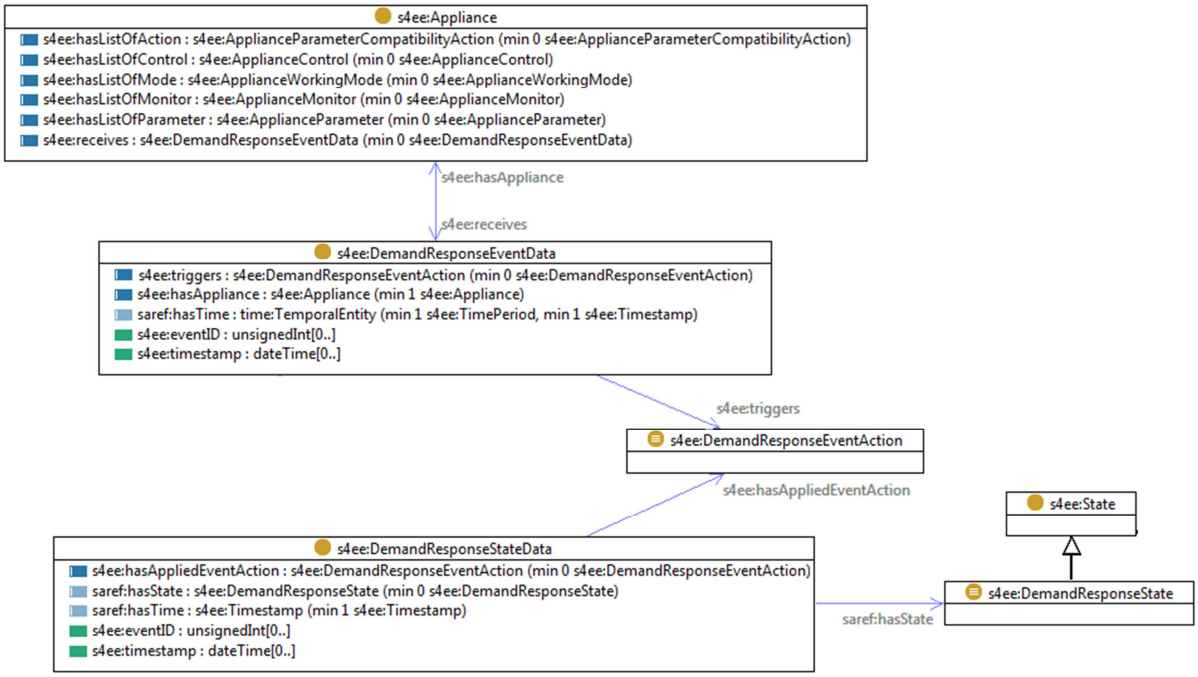

Figure 13 – Demand Response Event classes and properties

The s4ee:DemandResponseEventData class is used to represent overload warning severity level and related load control commands to an appliance. It is characterized by an event ID and a timestamp (which can be expressed either as a s4ee:timestamp datatype property or using the saref:hasTime object property).

The s4ee:DemandResponseEventAction class expresses the type of actions to be performed as a consequence of a demand response event. The type of actions can be s4ee:emergency, s4ee:increase, s4ee:normal, s4ee:pause, s4ee:reduce, s4ee:resume.

The s4ee:DemandResponseStateData class expresses the data about the state of an event and is characterized by the same event ID used in the s4ee:DemandResponseEventData class, as well as a timestamp, and it is associated to the class s4ee:DemandResponseState , which expresses the possible states of a demand response event, i.e., s4ee:eventAccepted, s4ee:eventStarted, s4ee:eventStopped, s4ee:eventRejected, s4ee:eventCancelled, or s4ee:eventError.