

# VU Research Portal

## Validating SAREF in a Smart Home Environment

van der Weerdt, Roderick; de Boer, Victor; Daniele, Laura; Nouwt, Barry

### **published in**

Metadata and Semantic Research  
2021

### **DOI (link to publisher)**

[10.1007/978-3-030-71903-6\\_4](https://doi.org/10.1007/978-3-030-71903-6_4)

### **document version**

Publisher's PDF, also known as Version of record

### **document license**

Article 25fa Dutch Copyright Act

[Link to publication in VU Research Portal](#)

### **citation for published version (APA)**

van der Weerdt, R., de Boer, V., Daniele, L., & Nouwt, B. (2021). Validating SAREF in a Smart Home Environment. In E. Garoufallou, & M-A. Ovalle-Perandones (Eds.), *Metadata and Semantic Research: 14th International Conference, MTSR 2020, Madrid, Spain, December 2–4, 2020, Revised Selected Papers* (pp. 35-46). (Communications in Computer and Information Science; Vol. 1355 CCIS). Springer Science and Business Media Deutschland GmbH. [https://doi.org/10.1007/978-3-030-71903-6\\_4](https://doi.org/10.1007/978-3-030-71903-6_4)

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

### **Take down policy**





If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

### **E-mail address:**

[vuresearchportal.ub@vu.nl](mailto:vuresearchportal.ub@vu.nl)



# Validating SAREF in a Smart Home Environment

Roderick van der Weerd<sup>1</sup> , Victor de Boer<sup>1</sup> , Laura Daniele<sup>2</sup> ,  
and Barry Nouwt<sup>2</sup> 

<sup>1</sup> Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

{r.p.vander.weerdt,v.de.boer}@vu.nl

<sup>2</sup> TNO - Netherlands Organization for Applied Scientific Research,  
The Hague, The Netherlands

{laura.daniele,barry.nouwt}@tno.nl

**Abstract.** SAREF is an ontology created to enable interoperability between smart devices. While the IoT community has shown interest and understanding of SAREF as a means for interoperability, there is a lack in the literature of practical examples to implement SAREF in real applications. In order to validate the practical implementation of SAREF we perform two experiments. First we map IoT data available in a smart home into RDF using SAREF. In the second part of the paper an IoT environment is created by using the Knowledge Engine, a framework created to allow communication between smart devices, operating on Raspberry Pi's emulating IoT devices, where the communication of the IoT devices is performed by sharing knowledge represented with SAREF. These experiments demonstrate that SAREF is an ontology that is successfully applicable in different situations, with data-mapping showing that SAREF is able to represent the information of different smart devices and by using the Knowledge Engine showing that SAREF can enable interoperability between smart devices.

**Keywords:** IoT · Ontology · Data mapping · Smart home.

## 1 Introduction

Over the last years more and more companies have started making smart devices that can be connected to the Internet of Things (IoT). Connecting multiple devices brings the promise that all those devices are now able to interact with each other in meaningful ways. Unfortunately the reality is that devices are not nearly as interconnected as one would want [1], often devices can only be accessed from specific (vendor-based) apps, other times the devices are not able to communicate because they do not speak the same language. In order to make the interoperability between smart devices within IoT possible the Smart Applications REference ontology (SAREF) has been created [2], an ontology specifically designed to encompass the information that smart devices need to exchange in order to have meaningful interactions.

© Springer Nature Switzerland AG 2021

E. Garoufallou and M.-A. Ovalle-Perandones (Eds.): MTSR 2020, CCIS 1355, pp. 35–46, 2021.

[https://doi.org/10.1007/978-3-030-71903-6\\_4](https://doi.org/10.1007/978-3-030-71903-6_4)

SAREF has been validated before [3,4] to determine the quality of the ontology. But until now there has not been a practical implementation of the SAREF ontology, to validate if it satisfies requirements needed to map real data from real devices. The goal of this paper is to create an example of said implementation, validating the ontology in two ways, its ability to express all the information available from multiple smart devices in a home and its ability to enable interoperability by representing messages in a meaningful manner allowing for communication between smart devices.

To achieve the first goal we map data collected in a smart home (Sect. 3) into RDF using SAREF. For the second goal the Knowledge Engine is proposed to validate the effectiveness of using SAREF for communication (Sect. 4). The paper will start with describing the SAREF ontology and related research.

## 2 Related Work

To provide a background to our experiments we will first present an in depth examination of the SAREF ontology, followed by three related ontologies and lastly other validation studies using competency questions to validate ontologies.

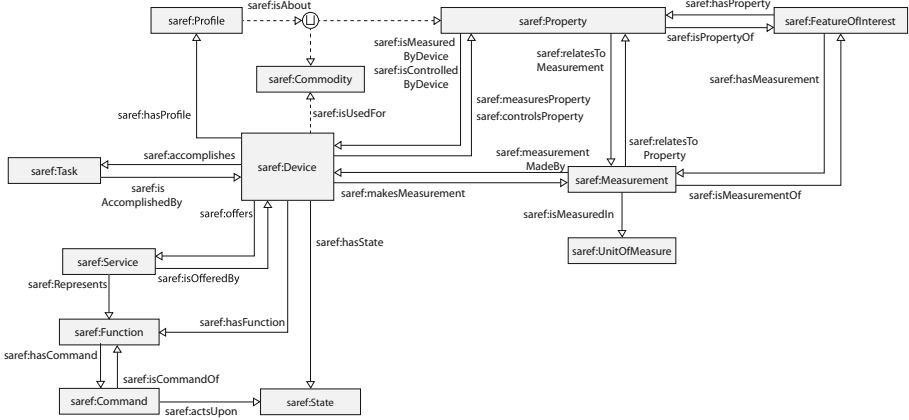
### 2.1 SAREF

The Smart Applications REference ontology (SAREF) is a reference ontology for IoT applications (<https://saref.etsi.org/>) [5]. Its development started in 2013, when the European Commission, in collaboration with the European Telecommunication Standardization Institute (ETSI), launched an initiative to build a common ontology in close collaboration with the smart appliances industry<sup>1</sup> [2]. In a fragmented landscape of IoT standards, platforms and technologies across different vertical domains [6,7], SAREF was created as a shared model of consensus that could enable the communication among various IoT devices from different manufacturers that use different protocols and standards [8]. The first proof-of-concept solution based on SAREF was demonstrated and implemented in 2017 on existing commercial products in the energy domain<sup>2</sup> [4]. SAREF is published as a series of ETSI technical specifications, consisting of a modular framework that comprises a generic core ontology for IoT [5] and 10 domain-specific extensions (ETSI TS 103 410, parts 1–10), such as SAREF for Energy, Buildings and Cities, amongst others. The SAREF framework is maintained and evolved by ETSI and experts from several European organizations that successfully collaborate with each other<sup>3</sup>. One of the latest supported initiatives is the

<sup>1</sup> <https://ec.europa.eu/digital-single-market/en/blog/new-standard-smart-appliances-smart-home>.

<sup>2</sup> <https://ec.europa.eu/digital-single-market/en/news/digitalising-energy-sector-common-language-consumer-centric-world>.

<sup>3</sup> See, for example, <https://portal.etsi.org/STF/STFs/STF-HomePages/STF534>, <https://portal.etsi.org/STF/STFs/STF-HomePages/STF566> and <https://portal.etsi.org/STF/STFs/STF-HomePages/STF578>.



**Fig. 1.** Overview of the SAREF ontology (Image taken from: <https://saref.etsi.org/core/v3.1.1/#Figure.1>).

development of an open portal for the SAREF community and industry stakeholders, so that they can contribute directly to the SAREF evolution (<https://forge.etsi.org/rep/SAREF>).

Figure 1 shows an overview of the main classes and relationships in SAREF. In total SAREF contains 81 classes, 35 object properties and 5 data properties. The starting point is the concept of *Device*, which is defined as a tangible object designed to accomplish a particular *Task*. In order to accomplish this task, the device performs a *Function*. For example, a temperature sensor is a device of type `saref:Sensor`, is designed for tasks such as `saref:Comfort`, `saref:WellBeing` or `saref:EnergyEfficiency`, and performs a `saref:SensingFunction`. Functions have commands. A *Command* is a directive that a device needs to support to perform a certain function. Depending on the function(s) it performs, a device can be found in a corresponding *State*. A device that wants (a certain set of) its function(s) to be discoverable, registerable, and remotely controllable by other devices in the network can expose these functions as a *Service*. A device can also have a *Profile*, which is a specification to collect information about a certain *Property* or *Commodity* (e.g. Energy or Water) for optimizing their usage in the home/building in which the device is located. A *Property* is defined as anything that can be sensed, measured or controlled by a device, and is associated to measurements. For example, a temperature sensor measures a property of type `saref:Temperature`. A *Measurement* is the measured value made over a property and must be associated to an unit of measure and a timestamp. The *Feature of Interest* concept further allows to represent the context of a measurement, i.e., any real world entity from which a property is measured. For example, whether the measured temperature is that of a room or of a person. A more detailed description of the SAREF classes and properties can be found in [5].

## 2.2 Related Ontologies

The Semantic Sensor Network (SSN) ontology, was specifically created for the modeling of sensors and their observations[9]. This strict focus on sensors allows for a modular use of the ontology with other ontologies. A sensor is defined as “anything that observes”, which means it does not include all the different kinds of smart devices that a smart home offers. The Sensor, Observation, Sample, and Actuator (SOSA) ontology is an extension and partial replacement of SSN created to model “the interaction between the entities involved in the acts of observation, actuation, and sampling” [10]. Because SSN was considered to rigid SOSA was created to be more flexible, having 13 concepts and 21 properties compared to the 41 concepts and 39 object properties of SSN.

The Web of Things Thing Description (TD) is a newer ontology that was also created by W3C [11]. This ontology models the metadata of the devices instead of the measurements. The creators envision that every smart device will its own TD to serve as identification and as a starting point for interaction with the device, similar to a index page of a website. But it lacks the representation of the measurements made by the devices that is needed for our experiments.

## 2.3 Competency Questions for Validation

Validation of an ontology can be done by creating a scenario and formulating a set of competency questions. Competency questions are proposed as a way to informally define requirements for an ontology, the resulting ontology should be able to answer all of the competency questions to prove it is the correct ontology to use in this scenario [12].

Competency questions have been used to validate SAREF before, as demonstrated in the research by Moreira et al. [4]. In their work they created a model that maps a knowledge graph using the SSN ontology into a knowledge graph using the SAREF ontology. Using an example situation of a wind sensor represented using the SSN ontology they were able to map the data to a knowledge graph, but were not able to keep all of the information that was available in the original representation. Competency questions were used to define the requirements that the ontologies required and showed that the new knowledge graph did not retain enough information from the original. The experiment they performed used an example specifically created for SSN, in this paper we will use data that comes from smart devices in a home setting, which is what SAREF was developed for and should therefore allow a better evaluation of SAREF.

Another example of competency questions being used as a tool to select different ontologies can be found in [3]. Bajaj et al. use the competency questions they define to determine which classes an ontology requires to fit to their requirements. For example SAREF was rejected by the authors as a suitable ontology because it lacked a class to define the concept of location. Since then a new addition to the SAREF ontology has been the Feature of Interest class, which allows for exactly such a definition.

Sagar et al. [13] extend the SOSA/SSN ontology in order to allow for smart sensors, sensors that are able to select and run different algorithms resulting in different data, to be modelled accurately. They create 13 competency questions and based on those reject existing ontologies, including SAREF. The S3N ontology that they propose was able to resolve all the competency questions. Instead of trying to model the algorithms behind “smarter” observations the decision was made to only look at the output of the devices for our experiment since this is the relevant information that we have available. Similarly we only look at the output of a thermometer and not at the intricate ways the device measures the temperature of the room.

### 3 Mapping Data Sources

The main purpose of SAREF is to enable interoperability between different devices and services [5]. In order to validate SAREF this experiment will focus on whether SAREF is capable of expressing all of the information from a smart home by mapping the smart devices to a knowledge graph using SAREF.

#### 3.1 Method

This experiment will take data from 33 smart devices in a smart home and transforms it into a knowledge graph using the SAREF ontology. The smart home is part of a pilot from a large construction company experimenting with adding smart devices to the houses they build. We started with removing duplicate smart devices that were used multiple times in the house such as the thermometer in the living room and in the bedroom. This resulted in nine data sources that will be used for this experiment:

- *current temperature*, measurement of the room temperature in degrees Celsius.
- *CO2 levels*, measurement of the ppm CO2 in a room.
- *last time movement detected*, marks the most recent moment (timestamp) that movement was detected in a room. Used to determine when residents are at home.
- *first time movement detected*, marks the beginning (timestamp) of new movement detected in a room. Used to determine when residents are at home.
- *motion detected*, returns a yes or no state based on whether the sensor detected motion.
- *presence detected*, returns a yes or no state based on whether the sensor detected presence, based on the CO2 levels in a room.
- *humidity*, measurement of the percentage of humidity in a room.
- *smart switch actuator*, an actuator that can be turned on or off by sending it an on/off command.
- *target temperature*, the desired temperature of a room in degrees Celsius, controlled by the thermostat.

All the sensors send updated information every five seconds, which is recorded with a timestamp. The complexity of the data from the sensors differs from the “simpler” *thermometer* to the more elaborate *last time movement detected*. As was discussed in Sect. 2.3 we only focus on the output of the device and not the internal calculations.

The combination of these data sources can be used in multiple scenarios, a possible scenario would be to adjust the room temperature when there are no people present. Using a combination of the *first time movement detected* data and the *last time movement detected* data, which in turn is based on the *motion detected* and *presence detected* data, we can determine whether a home is occupied. When there are no people present the target temperature could be set to a lower setting, meaning lower as opposed to the target temperature when there are people present. When the current temperature is below the target temperature a command can be sent to the heating system of the home (in the case that one of the smart switch actuators is connected to a heating system) shutting it down, this would allow a home to only be heated when it is needed.

### 3.2 Implementation

Not all classes shown in Fig. 1 are used in the mapping for a data source. For example a command that is sent to a smart actuator will not be using the `saref:Measurement` class. The Technical Specification [5] describing the SAREF ontology is used to make a selection of relevant classes for each smart device. The relevant classes for each device can be seen in Table 1. Most mappings are easily performed by selecting the straightforward SAREF class, for example for the `saref:Property` class in the mapping of the thermometer we can use its subclass `saref:Temperature` to mean that the measurement in this graph pattern relates to temperature.

For the sake of clarity this paper only contains mappings for two data sources and a summary of the special cases where the decisions made need an explanation. The complete mapping of all the data sources can be found online<sup>4</sup> where all the mappings are represented as triples.

**In Depth Mapping Explanations.** The two examples that are chosen to be clarified are the data from the thermometer and the data from the CO2 sensor. The thermometer mapping is an example of a mapping where all the relevant classes are already available. For the CO2 sensor two new subclasses have to be added. The OM1.8: Units of Measure ontology [14] was chosen to use to represent the units of measure, as suggested in [5].

**Thermometer.** The thermometer makes a measurement of the temperature in a room. The mapping requires the:

- `saref:Measurement` class for the value of the temperature and timestamp.

<sup>4</sup> [https://github.com/RoderickvanderWeerd/IoT\\_data\\_SAREF\\_mappings](https://github.com/RoderickvanderWeerd/IoT_data_SAREF_mappings).

**Table 1.** Overview of the classes used to model each data source.

	ct	Cl	ftmd	ltmd	md	pd	h	ssa	tt
saref:Measurement	x	x					x		x
saref:UnitOfMeasure	x	x					x		
saref:Property	x	x	x	x	x	x	x	x	x
saref:FeatureOfInterest	x	x	x	x	x	x	x	x	x
saref:Device	x	x	x	x	x	x	x	x	x
saref:Commodity									
saref:Profile									
saref:Task	x	x	x	x	x	x	x	x	x
saref:Service								x	
saref:Function	x	x	x	x	x	x	x	x	x
saref:Command								x	
saref:State			x	x	x	x		x	

**legend** ct: *current temperature*, Cl: *CO2 levels*, ftmd: *first time movement detected*, ltmd: *last time movement detected*, md: *motion detected*, pd: *presence detected*, h: *humidity*, ssa: *smart switch actuator*, tt: *target temperature*.

- **saref:UnitOfMeasure** class for the unit of the measurement, we can use the subclass **saref:TemperatureUnit** and **om:degree\_Celcius** as its instance.
- **saref:Property** class to map what the measurement is measuring, in this case temperature, so we can use its subclass **saref:Temperature**.
- **saref:FeatureOfInterest** class represents the feature of interest, in this case the room of which the measurements are made. The URI would represent the name of the room.
- **saref:Device** class to represent the device itself that is making the measurements, we can use the subclass **saref:Sensor**.
- **saref:Task** class to represent the goal of **saref:Comfort** for the user.
- **saref:Function** class to define the function as a **saref:SensingFunction**.

**CO2 sensor.** The CO2 sensor makes a measurement of the CO2 levels in a room. The mapping requires the:

- **saref:Measurement** class for the CO2 levels and the timestamp.
- **saref:UnitOfMeasure**, SAREF does not have a unit of measure subclass for CO2, a new subclass of UnitOfMeasure is created to represent CO2 measurements and the parts per million unit of OM1.8 was used as unit instance, resulting in the additional triple:  
`om:parts_per_million rdf:type ex:CO2Unit`
- **saref:Property**, in this case CO2 levels, for which a class is not available in SAREF, but can be easily added:  
`<PROPERTY_URI> rdf:type ex:CO2`
- **saref:FeatureOfInterest**, similar to the thermometer mapping.



- `saref:Device`, similar to the thermometer mapping.
- `saref:Task` class to represent the goal of the `saref:WellBeing` of the user.
- `saref:Function`, similar to the thermometer mapping.

Aside from the two cases described above there was one other case where extra work was required. Just like with the CO2 levels measurements the humidity measurements requires a new subclass for `saref:UnitOfMeasure` to express that it is a humidity measure. It uses the OM1.8 instance for percentage:

```
om:percent rdf:type ex:HumidityUnit
```

## 4 Sharing Data with SAREF

This part of the paper is designed to test the capability of SAREF to enable interoperability between different devices. First the scenario will be detailed and the competency questions will be formulated accordingly. To achieve interoperability, multiple separate devices should be setup that 1) do not have direct or hard links between them and 2) communicate exclusively using SAREF. To the best of our knowledge no other ontology-based interoperability framework in smart homes allows for the data to remain at the source where it is produced. Our findings should generalize to other interoperability frameworks.

The Knowledge Engine will be shortly described in Sect. 4.2, followed by the implementation and its results.

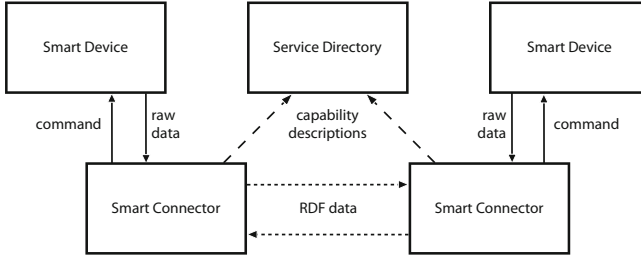
### 4.1 Competency Questions

For the sake of simplicity, the paper uses an simple scenario of a temperature controlled room, meaning we will need: information about what room the information is about, information about the temperature of the room, information about the desired temperature and the possibility to control the temperature of the room. Although more competency questions can be defined based on this scenario, we define here the following three competency questions as an example:

1. What is the temperature of a specific room?
2. Is this higher or lower than the threshold temperature?
3. Should the heater be turned on or off?

### 4.2 Knowledge Engine

The Knowledge Engine is a framework that allows multiple IoT devices to exchange knowledge in an interoperable way within a network. It is used by adding a smart connector to each device and registering this smart connector in the service directory, a visualization can be found in Fig. 2. The smart connector provides two functions to achieve interoperability: translation and discovery. It *translates* between a common ontology and the specific language of the device it is connected to, allowing devices from different vendors to understand each



**Fig. 2.** Visualization of the data flow in the Knowledge Engine.

other. It also dynamically *discovers* other smart connectors that can supply relevant data, which prevents hard links between devices. These two functions of the smart connector allow connections between devices in a network to be solely based on concepts and relations from a common ontology. This means that any device in the network can potentially be replaced by a similar one from a different vendor without loss of function i.e. they are interoperable with each other.

Both the translation and the discovery functions of the smart connector require configuration. Regarding the translation, this configuration consists of custom code provided by a developer that maps the specific device language to the common ontology. For discovery, the smart connector needs to be configured with the capabilities of the device (i.e. the *knowledge demand* and *knowledge supply*). The demand describes (in terms of the common ontology) the data that the device requires to function, while the supply describes the data that the device can provide to other devices. These capability descriptions use a SPARQL-like syntax to describe graph patterns, see Fig. 3. Apart from this configuration, the discovery function also requires a component called “Service Directory” to which all smart connectors in the network register themselves and from which they can retrieve the other smart connectors currently available.

A single smart connector is an extended version of the Apache Jena Fuseki triple store that is not used for actually storing the triples, but for its reasoner to orchestrate the knowledge exchange process. This reasoner uses rules that are based on the configuration (i.e. capabilities) of all available smart connectors in the network about which every smart connector regularly retrieves updates from the service directory. Whenever a device publishes or requests data, the smart connectors make sure that it is received by or from the correct device. Mutually, the smart connectors use a combination of SPARQL and a publish/subscribe mechanism to communicate.

For example, a thermometer device would have a supply capability description, like the one shown in Fig. 3b. The thermostat requesting this information uses a demand capability description that looks identical, because both supply and demand capability descriptions follow the same SPARQL-like structure, multiple triples with variables for the data that it either demands or supplies.

The Knowledge Engine can use any ontology to structure the knowledge that it shares (both capability descriptions and RDF data). For this experiment

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@prefix sosa: <http://www.w3.org/ns/sosa/>
@prefix saref: <https://saref.etsi.org/core/>
@prefix ex: <http://example.org/>

?id rdf:type ex:Room .
ex:Room rdfs:subClassOf saref:FeatureOfInt...
?id ex:hasName "Room1" .
?act saref:hasFeatureOfInterest ?id .
?act saref:actsUpon saref:OnOffState .
?act rdf:type ?command .

?id rdf:type ex:Room .
ex:Room rdfs:subClassOf saref:FeatureOfInterest .
?id ex:hasName "Room1" .
?obs rdf:type saref:Measurement .
?obs saref:isMeasurementOf ?id .
?obs saref:isMeasuredIn saref:TemperatureUnit .
?obs saref:hasValue ?temp .

```

(a) (b)

**Fig. 3.** Two capability descriptions that can be used with the Knowledge Engine.

SAREF was used, but it could also work with different ontologies like SOSA/SSN [10]. By using the Knowledge Engine we can validate SAREF by implementing it for the knowledge that is exchanged within its communications.

### 4.3 Implementation

The smart connectors run on Raspberry Pi 4B's, allowing easy configuration relevant to the device it is connected to with a config file<sup>5</sup>. The configuration requires a name and short description of the device and allows for inclusion of their capability descriptions. For this experiment, the smart device and smart connector are on one device with the devices being controlled with a python script that “makes” the devices smart allowing for communication with the Apache Fuseki server. A separate laptop runs the service directory and lastly, a WiFi router allows the smart connectors to connect to each other and to the service directory, as represented by the dashed lines in Fig. 2.

The first Raspberry Pi functions as a thermometer. It has one knowledge supply containing the temperature measurements made by the connected thermometer and the room it is located in. This knowledge supply can be found in Fig. 3b, `?room` is the name of the room and `?temp` is the latest measurement of the temperature.

The second Raspberry Pi functions as a thermostat, it has an internal desired state for the temperature of this room which can be adjusted with the connected buttons. It demands the temperature of this room (knowledge demand in Fig. 3b) and supplies a command based on that current temperature (knowledge supply in Fig. 3a), if the current temperature is below the desired state the smart connector will send `saref:OnCommand` for `?command`, if it is higher it will send `saref:OffCommand`.

The last Raspberry Pi represents a heating device, demanding a command for this room (knowledge demand in Fig. 3a). When the smart connector receives a `saref:OnCommand` it will turn on the heater (in this case the led light will turn on) and when it receives a `saref:OffCommand` it will turn it off.

<sup>5</sup> <https://jena.apache.org/documentation/fuseki2/fuseki-configuration.html>.

```

:Room1 rdf:type ex:Room .
ex:Room rdfs:subClassOf saref:FeatureOfInt..
:Room1 ex:hasName "Room1" .
:act1 saref:hasFeatureOfInterest :Room1 .
:act1 saref:actsUpon saref:OnOffState .
:act1 rdf:type saref:OnCommand .

:Room1 rdf:type ex:Room .
ex:Room rdfs:subClassOf saref:FeatureOfInterest .
:Room1 ex:hasName "Room1" .
:obs1 rdf:type saref:Measurement .
:obs1 saref:isMeasurementOf :Room1 .
:obs1 saref:isMeasuredIn saref:TemperatureUnit .
:obs1 saref:hasValue "17.0" .

```

(a)

(b)

**Fig. 4.** Example of a graph pattern from the thermostat smart connector (a) and the thermometer smart connector (b) answering the competency questions.

When all three of these smart connectors communicate correctly they will be able to control the temperature in a room. Smart connector b will receive the temperature of the room from smart connector a, compare it with the desired temperature (which can be adjusted with its buttons) and based on that send either an on or off command towards smart connector c.

#### 4.4 Answering the Competency Questions

In order to make all of the knowledge exchanges between the smart connectors possible two different capability descriptions are required, shown in Fig. 3.

Using the information from these devices we are able to answer all three of the competency questions. The graph patterns that are sent by the smart connector connected to the thermometer (Fig. 4b) answers the first question, the temperature of the room. The second competency question is answered internally in the smart connector connected to the thermostat and the third competency question is solved by the thermostat smart connector sending its graph pattern that is received by the third smart connector that interprets and either turns the heater off or, in the case of Fig. 4a, on.

## 5 Conclusion

In the first part of the paper we validated how data sources of a smart home scenario could be mapped to SAREF. We demonstrated that this can be done in a straightforward way. In the two cases where there were exceptions, as shown in Sect. 3.2 they were simple to add given the flexibility by design of SAREF. On the other hand, this could potentially lead to incompatibility issues if different people create their own extensions for similar situations. To that end, ETSI has recently created an open community portal ([saref.etsi.org](http://saref.etsi.org)) where SAREF users can submit their contributions that will then be considered in the official standardization process to become part of new SAREF releases.

The second part of the paper showed a simple, practical implementation of SAREF using the Knowledge Engine, which is a promising framework that allows multiple IoT devices to exchange knowledge in an interoperable way within a

network. The example used a scenario in which three devices shared knowledge to control the temperature of a room.

We acknowledge that the scenario presented in the paper is rather simple and more work in the immediate future is aimed at increasing the number and type of sensors in a more complex scenario than only heating the room, also considering more than one household. More future work could recreate the scenario we presented with other interoperability frameworks or using different ontologies.

**Acknowledgements.** This work is part of the Interconnect project (interconnect-project.eu/) which has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 857237.

## References

1. Hsu, C.L., Lin, J.C.C.: An empirical examination of consumer adoption of internet of things services: network externalities and concern for information privacy perspectives. *Comput. Human Behav.* **62**, 516–527 (2016)
2. Daniele, L., den Hartog, F., Roes, J.: Created in close interaction with the industry: the Smart Appliances REFERENCE (SAREF) ontology. In: Cuel, R., Young, R. (eds.) FOMI 2015. LNBIP, vol. 225, pp. 100–112. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-21545-7\\_9](https://doi.org/10.1007/978-3-319-21545-7_9)
3. Bajaj, G., Agarwal, R., Singh, P., Georgantas, N., Issarny, V.: A Study of Existing Ontologies in the IoT-Domain (2017). arXiv preprint: [arXiv:1707.00112](https://arxiv.org/abs/1707.00112)
4. Moreira, J., et al.: Towards IoT Platforms’ Integration Semantic Translations between W3C SSN and ETSI SAREF. In: SEMANTICS Workshops (2017)
5. ETSI TS: 103 264 V3.1.1, SartM2M; Smart Applications; Reference Ontology and oneM2M Mapping (2020)
6. ETSI TS: 103 375: SmartM2M; IoT Standards Landscape and Future Evolutions (2016)
7. ETSI TS: 103 376: SmartM2M; IoT LSP Use Cases and Standards Gaps (2016)
8. Daniele, L., Strabbing, W., Roelofsen, B., Aalberts, A., Stapersma, P.: Study on Ensuring Interoperability for Demand Side Flexibility. Technical report, European Commission (2018). <https://doi.org/10.2759/26799>
9. Compton, M., et al.: The SSN ontology of the W3C semantic sensor network incubator group. *J. Web Semant.* **17**, 25–32 (2012)
10. Janowicz, K., Haller, A., Cox, S.J., Le Phuoc, D., Lefrançois, M.: SOSA: a lightweight ontology for sensors, observations, samples, and actuators. *J. Web Semant.* **56**, 1–10 (2019)
11. W3C: Web of Things (WoT) Thing Description (2020). <https://www.w3.org/TR/2020/REC-wot-thing-description-20200409/>
12. Gruninger, M., Fox, M.: Methodology for the design and evaluation of ontologies. In: IJCAI 1995, Workshop on Basic Ontological Issues in Knowledge Sharing (1995)
13. Sagar, S., et al.: Modeling smart sensors on Top of SOSA/SSN and WoT TD with the semantic smart sensor network (S3N) modular ontology. In: 9th International Semantic Sensor Networks Workshop (2018)
14. Rijgersberg, H., van Assen, M., Top, J.: Ontology of units of measure and related concepts. *Semant. Web* **4**(1), 3–13 (2013)